XENIX® System V

Operating System

Operations Guide

This document was typeset with an IMAGEN® 8/300 Laser Printer.

SCO Document Number: XG-12-18-85-2.1/1.0

Contents

Chapter 1

Introduction

1.1 Overview

The XENIX operating system is a powerful system of programs, which allows you to accomplish a full spectrum of tasks, from developing high-level and assembly language programs to creating, editing, and typesetting documents. To keep this powerful system running smoothly, the XENIX system requires careful control of its operation and a regular schedule of maintenance. This guide explains how to operate and maintain the XENIX operating system on your computer, ensuring maximum performance with the least number of system problems.

This guide also explains how to expand a XENIX system with a Micnet network. A Micnet network allows serial communication between other XENIX systems in your work environment. The Micnet programs and commands include the **netutil** program, which is used to install the network, and the **mail**, **rcp**, and **remote** commands, which are used to pass messages, files, and commands over the network.

See Chapter 9, "Building A Micnet Network" for a complete explanation of this facility.

1.2 The System Manager

Every XENIX system should have at least one person in charge of system maintenance and operation. In this guide, that person is called the system manager. It is the system manager's duty to ensure the smooth operation of the system and to perform tasks that require special privileges.

Depending on the size of the system and the number of users on the system, a system manager's job can be anything from a once-a-week task to a full-time job. Even if the system is small, the system manager should faithfully perform each required maintenance task, since sloppy maintenance can affect XENIX performance.

All tasks in this guide are presented from the system manager's point of view, but many can also be accomplished by ordinary users. Since some of the tasks dramatically change the system's operation, we recommend that, whenever possible, the system manager perform these tasks. This can prevent unwanted or unnecessary changes to the system.

1.3 The Super-User Account

The super-user account is a special account for performing system maintenance tasks. It gives the system manager unusual privileges that ordinary users do not have, such as accessing all files in the system, and executing privileged commands. Many of the tasks presented in this guide require that the system manager be logged in as the super-user. To do this, the sys-

tem manager must know the super-user password created during the installation of the XENIX system (see the XENIX *Installation Guide*).

Users who are authorized to act as the super-user, including the system manager, should log in as the super-user only when it is necessary to perform a system maintenance task. Even if the system manager is the only person using the system, he should create a user account for himself and use it for day-to-day work, reserving the super-user account for system maintenance tasks only.

The number of individuals who are given the super-user password should be kept to a minimum. Misuse of the super-user powers by naive users can result in a loss of data, programs, and even the XENIX system itself.

1.4 The Keyboard

Many keys and key combinations have special meanings in the XENIX system. These keys and key combinations have special names that are unique to the XENIX system, and may or may not correspond to the keytop labels on your keyboard. To help you find the special keys, the following table shows which keys on a typical console correspond to XENIX system keys. A list for your particular console is in **keyboard**(M).

In this table, a hyphen (-) between keys means "hold down the first key while pressing the second."

XENIX Name	Keytop	Action
BREAK	Delete	Stops current action and returns to the shell. This key is also called the INTERRUPT or DELETE key.
BACKSPACE	Backspace	Deletes the first character to the left of the cursor.
Ctrl-D	Ctrl-D	Signals the end of input from the keyboard; also exits current shell or initiates the "logout" procedure if the current shell is the login shell.
Ctrl-H	Erase	Deletes the first character to the left of the cursor. Also called the ERASE key.

Ctrl-Q	Ctrl-Q	Restarts printing after it has been stopped with Ctrl-S.
Ctrl-S	Ctrl-S	Stops printing at the standard output device, for example a terminal. Does not stop the program.
Ctrl-U	Ctrl-U	Deletes all characters on the current line. Also called the KILL key.
Ctrl-\	Ctrl-\	Quits current command and creates a *core* file (Recommended for debugging only). See *core*(F) for more information.
ESCAPE	Esc	Exits the current mode; for example, exits insert mode when in the editor **vi**.
RETURN	Return	Terminates a command line and initiates an action from the shell.

Many of these special function keys can be modified by the user. See **stty**(C) for more information.

1.5 Using This Guide

The tasks presented in this guide range from simple ones requiring very little knowledge about XENIX, to quite complex tasks requiring extensive knowledge about XENIX and your computer.

Each chapter explains the tools and knowledge you need to complete the tasks described in that chapter. In some cases, you may be required to refer to other manuals, such as the XENIX *User's Guide*.

Chapter 1 introduces this guide.

Chapter 2 explains how to start and stop the XENIX system and how to log in as the super-user, the XENIX system's special system manager account.

Chapter 3 explains how to create accounts for the users who work on your system, how to assign groups, and how to manage user IDs.

Chapter 4 explains how to create and mount file systems, how to set permissions, and how to keep the system secure.

Chapter 5 explains how to maintain free space on the root file system and other file systems.

Chapter 6 explains how to create backup copies of the root file system and other file systems.

Chapter 7 explains how to add terminals, printers and other peripheral devices.

Chapter 8 explains how to solve system problems such as a jammed line-printer or a forgotten password.

Chapter 9 explains how to create a multiple system mailing network with Micnet.

Chapter 10 explains how to add device drivers to the XENIX system.

Appendix A presents a list of the XENIX system special files, and explains how to use these files when creating and maintaining file systems.

Appendix B presents a list of commonly used XENIX directories and log files.

Chapter 2
Starting and
Stopping the System

2.1 Introduction

This chapter explains how to start and stop the XENIX system. It also explains how to log in as the super-user.

2.2 Starting the System

Starting a XENIX system requires more than just turning on the power. You must also perform a series of steps to initialize the system for operation. Starting the system requires:

- Loading the operating system
- Cleaning the file system (if the system was improperly stopped)
- Choosing the mode of system operation

The following sections describe each of these procedures.

2.2.1 Loading the Operating System

The first step in starting the system is to load the operating system from the computer's hard disk. Follow these steps:

1. Turn on power to the computer and hard disk. The computer loads the XENIX bootstrap program and displays the message:

   ```
   Xenix boot
   :
   ```

2. Press the RETURN key. The bootstrap program loads the XENIX operating system.

When the system is loaded, it displays information about itself and checks to see if the "root file system" (i.e., all files and directories) is clean. If it is clean, you may choose the mode of operation. If not, the system requires you to clean the file system before choosing.

2.2.2 Cleaning the File System

You must clean the file system if the system displays the message:

```
Proceed with cleaning (y or n)?
```

This message indicates that the system was not stopped properly as described in the section, "Stopping the System", given later in this

chapter. The XENIX operating system requires a clean file system to perform its tasks.

To clean the file system, enter *y* (for "yes") and press the RETURN key. The system cleans the file system, repairing damaged files or deleting files that cannot be repaired. It reports on its progress as each step is completed. At some point, it may ask if you wish to salvage a file. Always answer by entering *y* and pressing the RETURN key.

When cleaning is complete, the system usually asks you to choose the mode of operation, but it may also display the message:

```
** Normal System Shutdown **
```

If it displays this message, you must reload the system. You can do this by resetting the computer and repeating the steps given in the previous section. For instructions on how to reset your computer, see the hardware manual provided with the computer.

2.2.3 Choosing the Mode of System Operation

You may choose the mode of XENIX operation as soon as you see the message:

```
Type CONTROL-d to continue with normal startup,
(or give the root password for system maintenance):
```

The system has two modes: *normal operation* and *system maintenance* mode. Normal operation is for ordinary work on the system. This is the mode you should choose to allow multiple users to log in and begin work. System maintenance mode is a specialized mode reserved for work to be done by the system manager. It does not allow multiple users.

To choose normal operation, press the Ctrl-D key. The system displays a startup message and begins to execute commands found in the command file */etc/rc* described in Chapter 8, "Solving System Problems." When the commands are finished, the system displays the "login:" prompt. You may then log in as a normal user, as decribed in the XENIX *Introduction to XENIX*, or as the super-user, as described in the next section.

To choose system maintenance mode, enter the super-user's password (sometimes called the "root password") and press the RETURN key. The system displays the message of the day and the maintenance mode prompt (#). The commands in the */etc/rc* file are not executed. (Choose system maintenance mode only if you must do system maintenance work that requires all other users to be off the system.)

2.3 Logging In As the Super-User

Many system maintenance tasks, when performed during normal operation, require that you log in as the super-user. For example, you must be logged in as the super-user to stop the system.

Before you may log in as the super-user, you need the super-user password. You also need to see the "login:" message on your terminal's screen. If you do not see this message, press the Ctrl-D key until it appears.

To log in as the super-user, follow these steps:

1. When you see the "login:" message, enter the super-user's login name

   root

   and press the RETURN key. The system prompts you for the super-user's password.

2. Enter the super-user's password and press the RETURN key. The system does not display the password as you enter it, so enter each letter carefully.

The system opens the super-user account and displays the message of the day and the super-user prompt (#).

Take reasonable care when you are logged in as the super-user. In particular, you should be very careful when deleting or modifying files or directories. Avoid using wildcard designators in filenames and frequently check your current working directory. Small errors can cause annoying and unwanted changes to the system and user files. Some errors can cause irretrievable damage to a file or the system.

You can leave the super-user account at any time by pressing Ctrl-D.

2.4 Stopping the System

Stopping the XENIX system takes more than just turning off the computer. You must prepare the system for stopping by using either the **shutdown** or the **haltsys** command. The following sections describe each command.

2.4.1 Using the shutdown Command

The **shutdown** command is the normal way to stop the system and should be used whenever the system is in normal operation mode. It warns other users that the system is about to be stopped and gives them an opportunity to finish their work.

To stop the system with the **shutdown** command, follow these steps:

1. Log in as the super-user (see the section "Logging in as Super-User" in this chapter). The system opens the super-user account and displays the message of the day and the super-user's prompt.

2. Enter:

   /etc/shutdown

   and press the RETURN key. The system loads the command which in turn prompts you for the number of minutes you wish to elapse before the computer stops:

   Minutes till shutdown? (0-15):

3. Enter any number from 0 to 15 and press the RETURN key. The system displays a warning message at each terminal, asking logged in users to finish their work and log out. As soon as all users are logged out or the specified time has elapsed, the system closes all accounts and displays the message:

   ** Normal System Shutdown **

   and stops.

You may now turn off the computer.

2.4.2 Using the haltsys Command

The **haltsys** command may be used to halt the system immediately. In general, it should be used only when no other users are on the system or when the system is in system maintenance mode.

To stop the system with the **haltsys** command, follow these steps:

1. Log in as the super-user (not required when in system maintenance mode). The system opens the super-user account and displays the message of the day and the super-user prompt.

2. Enter:

/etc/haltsys

and press the RETURN key. The system displays the message:

** Normal System Shutdown **

and stops.

You may now turn off the hard disk and computer.

Chapter 3

Preparing XENIX for Users

3.1 Introduction

User accounts help the XENIX system manager keep track of the people using the system, and control their access to the system's resources. Ideally, each user should have a user account. Each account has a unique "login name" and "password" with which the user enters the system, and a "home directory" where the user does his work.

It is the system manager's job to create accounts for all users on the system. It is also the manager's job to maintain user accounts by changing user passwords, login groups, and user IDs when necessary.

This chapter explains how to:

- Add user accounts to the system
- Change an account's password
- Force new passwords
- Create a group
- Change an account's login group
- Change an account's user ID
- Remove user accounts from the system

The following sections describe each in detail.

3.2 Adding a User Account

You may add a user account to the system with the **mkuser** program. The program creates a new entry in the XENIX system's */etc/passwd* file. This entry contains information about the new user, such as login name and initial password, that the system uses to let the user log in and begin work. The program also creates a home directory for the user, a mailbox for use with the **mail** command, and a *.profile* or *.login* file which contains XENIX commands that are executed when the user logs in.

To create a new user account, follow these steps:

1. Log in as the super-user.
2. Enter:

   mkuser

and press the RETURN key. The system displays the following message:

```
Newuser
-------
Add a user to the system

Do you require detailed instructions? (y/n):
```

3. Enter the letter *y* (for "yes"), if you want information about the program, otherwise type the letter *n* (for "no"). Enter *q* (for "quit") only if you wish to stop the program and return to the system. If you type a "q" to any "(y/n)" prompt, the program will stop and no changes will be made.

   When the program continues, it prompts you to enter the new user's login name:

   ```
   Enter new user's login name:
   ```

   The login name is the name by which XENIX will know the user. It is usually a short version of the user's actual name, typed in lowercase letters. For example, either "johnd" (a first name and last initial) or "jdoe" (a first initial and last name) is acceptable for the user John Doe.

4. Enter the new name, and press the RETURN key. The program now prompts you for information about the new user's group name and group number.

   A group name is the name of the group of users to which the new user will belong. Users in a group have access to a common set of files and directories. The group name is optional. If not given, the XENIX system's common group "group" (with group ID 50) is used.

   The program prompts:

   ```
   Do you want to use the default group? (y/n):
   ```

   If you enter "y", the user's group name will be "group" and the group ID number will be 50.

   If you enter "n", the program responds with a list of existing groups:

   ```
   Existing groups are:
   Group "group" (50): demo vdemo cdemo
   Do you want to use one of these groups? (y/n):
   ```

If you enter either "y" or "n", you are asked which group you want to use. Enter the name of the group. You may create a new group by entering in the new name.

Next, you are prompted for a group number. The group ID, or number, may be any number from 50 to 30000 that isn't already used for another group.

5. After entering the group name and ID, you are prompted for the initial password.

   Enter password:

   The initial password is the password you assign to the new user. The user will use the initial password to enter the account for the first time. Once in the account, the user should create a new password for himself, one that is hard to guess. (See the section "Changing Your Password" in Chapter 4 of the *Introduction to XENIX*.)

6. Enter the password, and press the RETURN key.

7. Next, you are prompted for a shell type. You see a list and brief explanation of the available shells (**sh**, **vsh**, **csh**, **rsh**, and **uucp login**) and the prompt:

   ENTER Shell type (1, 2, 3, 4 or 5) and press RETURN:

   sh is the standard (Bourne) shell. **vsh** is the menu driven "visual" shell, **csh** is the c-shell **rsh** is the restricted shell, and **uucp**login is an entry in */usr/lib/uucp/uucico* enabling logging in to the system via **uucp**.

8. Enter the desired shell number and press RETURN. After you have entered the shell type, the program prompts you for a comment:

   Please enter Comment >----------
   >

   A comment is information about the new user, such as a department name and phone extension. Although, the comment is optional, it is useful if the **finger** command is often used to display information about users. If given, the comment must be no more than 20 characters long, including spaces. It must not contain any colons (:). The example

   John Doe, 123

   shows the recommended form for a comment.

9. Enter the comment. Make sure it is 20 characters or less. If you do not wish to enter a comment, just press the RETURN key.

   The program now displays what you have entered and the special user entry that it has created for the new user. This entry is copied to the special system file */etc/passwd*. The entry shows the login name, the password (encrypted), the user ID, the group ID, the comment, the user's home directory, and the startup program. Items in the entry are separated by colons (:). (For a full description of each item, see *passwd*(M) in the XENIX *Reference Manual*.)

   The program then gives you an opportunity to change the user name, password, group, or comment:

   ```
   Username is "johnd", user ID is 2001.
   Group name is "group", group number is 50.
   Comment field is: "John Doe, 123"
   Shell is "/bin/csh"

   Do you want to change anything? (y/n):
   ```

10. Enter the letter *y* (for "yes") and press the RETURN key, if you wish to change something. Enter *n* (for "no") and skip to the next step if you wish to complete the new account. (Enter *q*, for "quit", only if you wish to leave the program and abort the new account.)

    If you enter *y*, the program prompts for the item you wish to change:

    ```
    username
    password
    group
    comment
    shell
    ```

    Enter the name of the item you wish to change and press RETURN. After you have changed an item, you see the complete list of items and are asked if you wish to make other changes. When you are finished with any changes, the program adds the user.

11. The program displays the message:

    ```
    Password file updated
    ```

    followed by a description of the actions it has taken to add the new user account to the system. The program then asks if you wish to add another user to the system.

12. Enter *y* if you wish to add another user. Otherwise, enter *n* to stop the program and return to the super-user prompt.

A user can log into a new account as soon as it is created. For details see the *Introduction to XENIX*.

3.3 Changing a User's Password

Normally, an ordinary user can change the password of his own account with the **passwd** command (see the *Introduction to XENIX*). Sometimes, however, it may be necessary for the super-user to change the password for him, for example, if the user has forgotten his password and cannot get into the account to change it. The super-user may change the password of any user (including himself) with the **passwd** command.

To change a password, follow these steps:

1. Log in as the super-user.
2. Enter:

   passwd *login-name*

   (where *login-name* is the user's login name) and press the RETURN key. The command displays the message:

   New password:
3. Enter the new password and press the RETURN key. The command does not display the password as you type it, so type carefully. The command then prompts you to enter the password again:

   Retype new password:
4. Enter the password again and press the RETURN key.

To see how an ordinary user can change his own password with the *passwd* command, see the *Introduction to XENIX*.

3.4 Forcing a New Password

From time to time, a user account may need a higher level of security than ordinary. Since the security of any account depends its password, it is important to keep the password as secret as possible. One way to provide greater security is to force users to change their passwords on a regular basis.

You can force users to change their passwords by using the **pwadmin** command. This command automatically dates each password and requires the user to provide a new password when the specified number of weeks have passed. The command also requires users to wait a minimum number of

weeks before allowing them to restore their previous password. To use the **pwadmin** command, you must log in as the super-user.

You can enable password aging for a specified user by using the **–a** option. Enter:

pwadmin –a *login-name*

where *login-name* is the login name of a user. The user will then be required to wait a minimum number of weeks before he can change his password, and will be forced to change his password after a maximum number of weeks have elapsed. The **–a** option uses the default minimum and maximum values found in the */etc/default/password* file.

You can choose your own minimum and maximum number of weeks by using the **–min** and **–max** options. For example, a common pair of minimum and maximum values is 2 and 8. To set the minimum and maximum dates, enter:

pwadmin -min *num* -max *num login-name*

where *num* is a number in the range 0 to 63, and *login-name* is simply the login name of the user whose password you are administering. Note that the minimum and maximum cannot both be 0, and that the minimum must not be greater than the maximum.

If you are unsure of the current minimum and maximum values for a password, you can display them by entering:

pwadmin -d *login-name*

This command does not change the current values.

If you wish to force a user to change his password immediately, enter:

pwadmin -f *login-name*

The user is prompted on his next login to supply a new password.

When a password no longer requires extra security, you can remove the current minimum and maximum values for the password by entering:

pwadmin -n *login-name*

The system will no longer prompt for changes.

For more information about password aging, see *pwadmin*(C) and *passwd*(M) in the XENIX *Reference Manual*.

3.5 Creating a Group

A group is a collection of users who share a common set of files and directories. The advantage of groups is that users who have a common interest in certain files and directories can share these files and directories without revealing them to others. Initially, all users belong to the common system group named "group", but you can create new groups by modifying the XENIX system file */etc/group* using a XENIX text editor.

To create a new group, you need to choose a group name and a group identification number (group ID). You also need to make a list of the users in the new group. The group name may be any sequence of letters and numbers up to eight characters long, and the group ID may be any number in the range 50 to 30000. Both the group name and ID must be unique, i.e., they must not be the same as any existing group name or ID.

To create a new group, follow these steps:

1. Log in as the super-user.

2. Display the contents of the */etc/group* file by entering:

   cat /etc/group

   and pressing the RETURN key. The **cat** command displays the contents of the */etc/group* file. The file contains several entries, each defining the group name, group ID, and users for a group. Each entry has the form:

   group-name::*group-ID*:*users*

   The users are shown as a list of login names separated by commas (,). For example, a typical file may look like this:

   ```
   other:x:1:demo
   sys:x:2:
   group::50:johnd,suex
   ```

3. Check the */etc/group* file entries to see that the group name and ID you have chosen are unique.

4. If the group name and ID are unique, invoke a XENIX text editor (see the XENIX *User's Guide*) and specify */etc/group* as the file to edit.

5. Locate the last line in the file, then insert the new entry in the form given above. For example, if you wish to create a group named "shipping" with group ID "142" and users "johnd", "marym", and "suex", enter:

```
shipping::142:johnd,marym,suex
```

6. Exit the editor.

To make sure you have entered the group names correctly, use the **grpcheck** command to check each entry in the */etc/group* file. If the new entry is free of errors, no other changes to the file are required.

You can create any number of new groups. Each group may have any number of members. Furthermore, any user may be a member of any number of groups. Multiple group membership is especially convenient for users who have interests that span a variety of areas.

If a user is a member of several groups, he can gain access to each group by using the **newgrp** command. See the *Introduction to XENIX* for details.

3.6 Changing a User's Login Group

When a user logs in, the system automatically places the user in his "login group". This is the group given by the group ID in the user's */etc/passwd* file entry (see the section "Adding a User Account" in this chapter). You can change the user's login group by changing the group ID. To change the group ID you need the group ID of the new login group, and you need to know how to use a XENIX text editor (see the XENIX *User's Guide*).

To change the group ID, follow these steps:

1. Log in as the super-user.

2. Use the **cd** command to change the current directory to the */etc* directory. Enter:

```
cd /etc
```

3. Use the **cp** command to make a copy of the */etc/passwd* file. Enter:

```
cp passwd passwd+
```

4. Invoke a text editor and specify */etc/passwd+* as the file to edit.

5. Locate the desired user's password entry. Each entry begins with the user's login name.

6. Locate the user's group ID number in the user's password entry. It is the fourth item in the entry. Items are separated by colons (:). For example, the entry:

   marym:9iKlwp:205:50:Mary March, 122:/usr/marym:/bin/sh

   has group ID "50".

7. Delete the old group ID and insert the new one. Be sure you do not delete any other portion of the user's password entry.

8. Exit the editor.

9. Use the **mv** command to save the old */etc/passwd* file. Enter:

   mv passwd passwd-

10. Use the **mv** command to make the edited file the new */etc/password* file. Enter:

    mv passwd+ passwd

You can make sure you have entered the new login group correctly by using the **pwcheck** command. If the new entry is correct, no other changes to the file are required.

You must not change the group IDs for system accounts such as "cron" and "root". System accounts are any accounts whose user IDs are less than 200. The user ID is the third item in the password entry.

Note that changing a user's login group does not change the "group ownership" of his files. Group ownership defines which group has access to a user's files. If users in the new group wish to access the user's files, you must change the group ownership with the **chgrp** (for "change group") command. For details, see the section "Changing Group Ownership" in Chapter 4.

3.7 Changing a User ID

Sometimes it is necessary to change the user ID in a user's account entry to allow a user to access files and directories transferred from other computers. In particular, if a user has different accounts on different computers and frequently transfers files and directories from one computer to another, the user IDs in each of his account entries must be made the same.

You can make them the same by modifying the account entries in the */etc/passwd* file.

To change a user ID, follow these steps at every computer for which the user has an account:

1. Log in as the super-user.

2. Use the **cd** command to change the current directory to the */etc* directory. Enter:

   ```
   cd /etc
   ```

3. Use the **cp** command to make a copy of the */etc/passwd* file. Enter:

   ```
   cp passwd passwd+
   ```

4. Invoke a XENIX text editor and specify */etc/passwd+* as the file to edit.

5. Locate the user's account entry. Each entry begins with the user's login name.

6. Locate the current user ID. The ID is the third item in the entry. For example, the entry:

   ```
   marym:9iKlwp:205:50:Mary March, 122:/usr/marym:/bin/sh
   ```

   has a user ID "205". Substitute the new user ID for the old one.

7. Exit the text editor.

8. Use the **mv** command to save the old */etc/passwd* file. Enter:

   ```
   mv passwd passwd-
   ```

9. Use the **mv** command to make the edited file the new */etc/passwd* file. Enter:

   ```
   mv passwd+ passwd
   ```

No other changes to the file are required.

In most cases, you can change the user ID to the same number as the user's most-used account. But the new number must be unique at every system for which the user has an account. If there is any conflict (for example, if the number already belongs to another user on one of the systems), you must choose a new number. You can choose any number greater than 200. Just make sure it is unique, and that you copy it to all systems on which the user has an account.

Once a user's ID has been changed, you must change the "user ownership" of the user's files and directories from the old user ID to the new one. You can do this with the **chown** (for "change owner") command described in Chapter 4, "Using File Systems."

For example, to change the ownership of johnd's home directory, enter:

```
chown johnd /usr/johnd
```

Note that you may use the **find** command described in Chapter 6, "Backing Up File Systems," to locate all files and directories with the user's old user ID.

3.8 Removing a User Account

It is sometimes necessary to remove a user account from the system. You can remove a user account with the **rmuser** program. The program deletes the user's entry from the */etc/passwd* file and removes the user's home directory and mailbox.

Before you can remove the user account, you must remove all files and directories from the user's home directory, or move them to other directories. If you wish to save the files, you may use the **tar** command to copy the files to a floppy disk (see the section "Copying Files to a tar disk" in Chapter 6).

To remove a user account, follow these steps:

1. Log in as the super-user.

2. Enter:

   cd /usr/*login-name*

   and press the RETURN key to change to the user's home directory. The *login-name* must be the user's login name.

3. Make sure that you have made copies of all important files and directories in the user's home directory.

4. Use the **rm** (for "remove") command to remove all files and directories from the user's home directory. This includes any files that begin with a period (.). Directories can be removed by using the **-r** (for "recursive") option of the **rm** command. For example, the command:

   ```
   rm -r bin
   ```

removes the directory named *bin* and all files within this directory.

5. After removing all files and directories, make sure the user's mailbox is empty. Enter:

   cat /usr/spool/mail/*login-name*

   and press the RETURN key, where *login-name* is the user's login name. If the mailbox contains text, enter:

   cat /dev/null > /usr/spool/mail/*login-name*

   and press the RETURN key.

6. When the user's home directory and mailbox are empty, enter:

   cd /usr

   and press the RETURN key. The user's home directory cannot be removed until you have moved to another directory.

7. Enter:

   rmuser

   and press the RETURN key. The program displays a message explaining how to remove a user:

   ****rmuser-remove a user from the system****

   Press ENTER when you are ready.

   The program then prompts you for the login name of the user you wish to remove:

   Enter name of id to be removed.

8. Enter the user's login name. You should now see the message:

   Removing user *name* from the system. CONFIRM? (y/n/q):

9. Enter *y* (for "yes") to remove the user from the system. Otherwise enter *n* (for "no") to stop the removal, or *q* (for "quit") to stop the program. The program removes the user's entry from the */etc/passwd* file, the user's mailbox, *.profile* file, and home directory. The program displays the message:

   User *name* removed from the system

The program now gives you a chance to remove another user:

Do you want to remove another user? (y/n/q):

10. Enter *y* to remove another user. Otherwise, enter *n* or *q* to stop the program.

Note that the **rmuser** program will refuse to remove an account that has a system name, such as "root", "sys", "sysinfo", "cron", "uucp", or a system ID (user ID below 200). Also, the program cannot remove a user account if the user's mailbox still has mail in it, or if the user's home directory contains files other than *.profile*.

Chapter 4

Using File Systems

4.1 Introduction

This chapter describes one of the most important responsibilities of a system manager: controlling and recording users' access to the files and directories on the system. It introduces file systems, permissions, system security, and process accounting.

4.2 File Systems

A file system is the XENIX system's way of organizing files on mass storage devices such as hard and floppy disks. A file system consists of files, directories, and the information needed to locate and access these items.

Each XENIX system has at least one file system. This file system is called the root file system and is represented by the symbol "/". The root file system contains all the XENIX programs that may be used by the system manager. It usually contains all the user directories as well.

A XENIX system may also have other file systems that contain user directories and application programs. One reason for using other file systems is to expand the available storage space of the system. Each additional file system adds its free space to the system's total space. New file systems can be specifically created by a user, then mounted onto the system so they can be used.

You can create a file system with the **mkfs** command. This command sets the size and format of the file system and may also copy some files to the new system. You can mount a file system with the **mount** command. Once mounted, you may access the files and directories in the file system as easily as files and directories in the root file system. (The root file system is permanently mounted.) When you are finished with a file system, you can unmount it with the **umount** command.

You can create new file systems on hard and floppy disks. A reason for creating new file systems on floppy disks is to establish a collection of application programs and data files that can be easily mounted and used when needed.

The following sections explain how to create and use file systems.

4.2.1 Creating a File System

You can create a file system on a formatted floppy disk by using the **mkfs** command.

To create the file system, you need:

- A formatted floppy disk
- The special filename of a floppy disk drive
- The disk block size of the disk
- The gap and block numbers for the disk

To format a floppy disk, see the section "Formatting Floppy Disks" in this chapter. The special filenames for the disk drives, the disk block size, and the gap and block numbers depend on the specific system and are given in Appendix A.

Note that if a file system already exists on the disk, it will be destroyed by this procedure. For this reason, be particularly careful not to create a new file system on the root file system. If you destroy the root file system, you will have to reinstall the XENIX system.

To make a file system on a floppy disk, follow these steps:

1. Log in. You do not have to be logged in as the super-user to use the **mkfs** command.

2. Insert a formatted floppy disk into a floppy disk drive. Make sure there is no read-only tab on the disk jacket.

3. Enter:

   /etc/mkfs *specialfile blocksize gap block*

   (where *specialfile*, *blocksize*, *gap*, and *block* are supplied by you) and press the RETURN key. The system automatically creates the file system. If it discovers data already on the disk, the system displays the message:

   mkfs: *specialfile* contains data. Overwrite? (y/n):

   If you are sure the disk contains nothing that you want to save, enter *y* and press the RETURN key to overwrite the data and continue creating the file system. Otherwise, enter *n*. If you enter *n*, no file system is created.

For example, the following command creates a file system on the floppy disk drive */dev/fd1*, with blocksize 320 and gap and block numbers 2 and 8:

```
/etc/mkfs /dev/fd1 320 2 8
```

The actual filename, blocksize, gap, and block numbers vary. For the information specific to your machine see Appendix A.

4.2.2 Mounting a File System

Once you have created a file system, you can mount it with the **mount** command. To mount a file system you need:

- The special filename of a disk drive
- The name of an empty directory

The special filenames of disk drives are given in Appendix A. The directory to receive the file system may be any directory as long as it is empty (contains no files) and is not your current working directory. Note that the directory */mnt* is specifically reserved for mounted file systems.

To mount a file system, follow these steps:

1. Log in. You do not have to be logged in as the super-user to use this command.

2. Insert the disk containing the file system into a floppy disk drive.

3. Enter the appropriate **mount** command, and press the RETURN key. The command should have the form:

   /etc/mount *specialfile directoryname*

   where *specialfile* is the special filename of the disk drive containing the disk, and *directoryname* is the name of the directory to receive the file system. If the disk has a read-only tab, make sure you include the switch **-r** at the end of the command.

For example, you can use the following command to mount the disk in disk drive */dev/fd1* onto the directory named */account:*

```
/etc/mount /dev/fd1 /account
```

Remember to make sure that the specified directory is empty before issuing the command. If the command displays the message:

```
mount: Structure needs cleaning
```

use the **fsck** command to clean the file system and try to mount it again (see the section "File System Integrity" in Chapter 5). If the command displays the message:

mount: Device busy

either the file system has already been mounted and cannot be mounted twice, or a user is currently in the directory. You must wait for users to leave a directory before you can mount the directory.

To check that the file system was properly mounted, use the **cd** command to change to the directory containing the mounted system and the **l** command to list the contents. The command displays the files and directories in the file system. Be sure to use the **cd** command to leave the directory after finishing your work in it.

Note that frequently used file systems can be mounted automatically when starting the system by appending the appropriate **mount** commands to the */etc/rc* file. See the section,"Changing the */etc/rc* File," in Chapter 8 for details.

4.2.3 Unmounting a File System

You can unmount a mounted file system with the **umount** command. Unmounting a file system does not destroy its contents. It merely removes access to the files and directories in the file system.

To unmount a mounted file system, enter:

/etc/umount *specialfile*

and press the RETURN key. The *specialfile* is the name of the special file corresponding to the disk drive containing the disk with the file system. The command empties the directory that previously contained the file system, and makes the directory and the corresponding disk drive available for mounting another file system.

For example, the following command unmounts a file system from the disk drive */dev/fd1*:

/etc/umount /dev/fd1

Before unmounting a file system, make sure that no files or directories are being accessed by other commands or programs. The **umount** command displays the following message if you or another user are currently in the directory containing the file system:

umount: Device busy

4.2.4 Formatting Floppy Disks

You can format floppy disks with the **format** program. Formatted disks are required whenever you create a file system. They are also required when you back up a file system with the **sysadmin** program (see Chapter 6, "File System Backups").

To format a floppy disk, follow these steps:

1. Insert a disk into floppy disk drive 0. Make sure there is no read-only tab on the disk jacket.

2. Enter:

   format /dev/device-name

   and press the RETURN key. The program formats the disk.

Warning: The format command is machine-specific. Refer to the manual page originally included in the XENIX *Installation*Guide for more information.

If for some reason you have not properly inserted the disk into the drive, the program displays an error message and stops.

In general, the system manager should format spare floppy disks in advance. Note that formatting removes all data from the disk, so if you are formatting a disk that already contains data, make sure that the data is nothing you wish to save.

4.3 Permissions

Permissions control access to all the files and directories in a XENIX system. In XENIX, ordinary users may access those files and directories for which they have permission. All other files and directories are inaccessible.

There are three different levels of permissions: user, group, and other. User permissions apply to the owner of the file; group permissions apply to users who have the same group ID as the owner; and other permissions apply to all other users.

4.3.1 Displaying Permissions

You can display the permission settings for all the files in a directory with the **l** (for "list directory") command. This command lists the permissions along with the name of the file's owner, the size (in bytes), and the date and time the file was last changed. The command display has the following format:

```
-rw-rw----1 johnd group 11515 Nov 17 14:21 file1
```

The permissions are shown as a sequence of ten characters at the beginning of the display. The sequence is divided into four fields. The first field (the "type" field) has a single character, the other fields ("user", "group", and "other") have three characters each. The characters in the fields have the following meanings.

In the type field:

d Indicates the item is a directory

\- Indicates the item is an ordinary file

b Indicates the item is a device special block I/O file

c Indicates the item is a device special character I/O file

In the user, "group", and "other" fields:

r Indicates read permission. Read permission for a file means you may copy or display the file. Read permission for a directory means you may display the files in that directory.

w Indicates write permission. Write permission for a file means you may change or modify the file. Write permission for a directory means you may create files or subdirectories within that directory.

x Indicates execute permission (for ordinary files) or search permission (for directories). Execute permission for a file means you may invoke the file as you would a program. Execute permission for a directory means you may enter that directory with the **cd** command.

\- Indicates no permission.

For example, the permissions:

```
-rwxrwxrwx
```

indicate an ordinary file with full read, write, and execute access for everyone (user, group, and other).

The permissions:

-rw-------

indicate an ordinary file with read and write access for the user only.

The permissions:

drwxr-x--x

indicate a directory with search access for everyone, read access for the user and group, and write access for only the user.

When you create a file, the XENIX system automatically assigns the following permissions:

-rw-r--r--

This means that everyone may read the file, but only the user may write to it. When you create a directory, the system assigns the permissions:

drwxr-xr-x

This means everyone may search and read the directory, but only the user may create and remove files and directories within it.

4.3.2 Changing Permissions

You can change the permissions of a file or a directory with the **chmod** (for "change mode") command. This command requires that you tell it how to change the permissions of a specific file or directory. You do so by indicating which levels of permissions you wish to change (user "u", group "g", or other "o"), how you wish to change them (add "+" or remove "-"), and which permissions you wish to change (read "r", write "w", or execute "x"). For example, the pattern:

u+x

adds execute permission for the user. The pattern:

go-w

removes write permission for group and other.

The chmod command has the form:

chmod *pattern file* ...

where *file* is the name of a file or directory. If more than one name is given, they must be separated by spaces. For example, to change the permissions of the file "receivables" from "-rw-r--r--" to "-rw-------", enter:

```
chmod go-r receivables
```

Press the RETURN key.

After using **chmod**, use the **l** command to check the results. If you have made a mistake, use **chmod** again to correct the mistake.

4.3.3 Changing the File Creation Mask

The file creation mask is a special number, kept by the system, that defines the permissions given to every file and directory created by a user. Initially, the mask has the value "022" which means every file receives the permissions:

```
-rw-r--r--
```

Every directory receives the permissions:

```
drwxr-xr-x
```

You can change the mask and the initial permissions your files and directories receive by using the **umask** command.

The **umask** command has the form:

```
umask value
```

where *value* is a three-digit number. The three digits represent user, group, and other permissions, respectively. The value of a digit defines which permission is given as shown by the following table:

Digit	Permission
0	Read and write (also execute for directories)
1	Read and write
2	Read (also execute for directories)
3	Read
4	Write (also execute for directories)
5	Write
6	Execute for directories
7	No permissions

For example, the command:

umask 177

sets the file creation mask so that all files and directories initially have read and write permission for the user, and no permissions for all others.

4.4 Managing File Ownership

Whenever a file is created by a user, the system automatically assigns "user ownership" of that file to that user. This allows the creator to access the file according to the "user" permissions. The system also assigns a "group ownership" to the file. The group ownership defines which group may access the file according to the "group" permissions. The group is the same group to which the user who created the file belongs.

Only one user and one group may have ownership of a file at any one time. (These are the owner and group displayed by the **l** command.) However, you may change the ownership of a file by using the **chown** and **chgrp** commands.

4.4.1 Changing User Ownership

You can change the user ownership of a file with the **chown** command. The command has the form:

chown *login-name file* ...

where *login-name* is the name of the new user, and *file* is the name of the file or directory to be changed. For example, the command:

chown johnd projects.june

changes the current owner of the file *projects.june* to "johnd".

The **chown** command is especially useful after changing the user ID of a user account (see the section, "Changing a User's ID" in Chapter 3).

You must be logged in as the super-user to use this command.

4.4.2 Changing Group Ownership

You can change the group ownership of a file with the **chgrp** command.

The command has the form:

chgrp *group-name file ...*

where *group-name* is the name of a group given in the */etc/group* file and *files* are the name of the file you wish to change. For example, the command:

chgrp shipping projects.june

changes the group ownership of the file *projects.june* to the group named "shipping".

The **chgrp** command is especially useful if you have changed the login group of a user (see the section, "Changing a User's Login Group" in Chapter 3).

4.5 System Security

Every system, no matter what its size, should have some form of protection from unauthorized access to the computer, disks, and system files. The following sections suggest ways for a system manager to protect the system.

4.5.1 Physical Security

You can protect the physical components of the computer, especially system disks, by taking these steps:

1. Keep unessential personnel out of the work area.

2. Organize and lock up all floppy disks when not in use. They should not be stored with the computer itself.

3. Keep disks away from magnetism, direct sunlight, and severe changes in temperature.

4. Do not use ball point pens to write labels on disks.

5. Make backup copies of all floppy disks (see the section, "Copying Floppy Disks," in this chapter).

4.5.2 Access Security

You can protect the system from access by unauthorized individuals by taking these steps:

1. Remind users to log out of their accounts before leaving the terminal.

2. Discourage users from choosing passwords that are easy to guess. Passwords should be at least six characters long and include letters, digits, and punctuation marks.

3. Keep the super-user password secret from all but necessary personnel.

4.5.3 Protecting Special Files

You can prevent ordinary users from gaining direct access to the data and program files on the system's hard and floppy disks by protecting the system's special files. The XENIX special files, in the */dev* directory, are used primarily by the system to transfer data to and from the computer's hard and floppy disks, as well as other devices, but can also be used by ordinary users to gain direct access to these devices.

Since direct access bypasses the system's normal protection mechanisms and allows ordinary users to examine and change all files in the system, it is wise to protect the special files to ensure system security.

To protect the XENIX special files, log in as the super-user and use the **chmod** command to set appropriate permissions. For example, to disallow any access by ordinary users, set the permissions of such special files as */dev/mem*, */dev/kmem*, */dev/root*, and */dev/usr* to read and write access for the user only. Note that you must not change the permissions for the */dev/tty* files.

4.5.4 Copying Floppy Disks

To ensure against the loss of data stored on floppy disks, you can use the **dd** command to make copies of floppy disks on new, formatted disks.

To make a copy of a disk, follow these steps:

1. Insert the disk to be copied into floppy drive 0.

2. Insert an empty, formatted disk into drive 1. If necessary, you can format a disk with the **format** command described in "Formatting Floppy Disks" in this chapter.

3. Enter:

   dd if=/dev/fd0 of=/dev/fd1 count=*blkcount*

   and press the RETURN key. The *blkcount* must be the number of blocks on the disk to be copied (see Appendix A for details).

The command copies the first disk to the second, then displays a record of the number of blocks copied.

4.6 Using XENIX Accounting Features

The XENIX system provides a set of commands that allow the system manager to perform process accounting. Process accounting is a simple way to keep track of the amount of time each user spends on the system. The process accounting commands keep a record of the number of processes (i.e., programs) started by a user, how long each process lasts, and other information such as how often the process accesses I/O devices, and how big the process is in bytes.

Process accounting is helpful on systems where users are being charged for their access time, but it may also be used to develop a detailed record of system, command, and system resource usage.

There are several commands which may be used to do process accounting. Of these, the most useful are **accton** and **acctcom**. The **accton** command starts and stops process accounting. When invoked, the command copies pertinent information about each process to the file named */usr/adm/pacct*. The **acctcom** command is used to display this information. The command has several options for displaying different types of accounting information.

4.6.1 Starting Process Accounting

Process accounting can be started at any time, but is typically started when the system itself is started. You can start process accounting with the **accton** command. Enter:

```
accton /usr/adm/pacct
```

The command automatically creates a new file */usr/adm/pacct* and begins to copy process accounting information to it. If the */usr/adm/pacct* file exists before starting **accton**, the file contents are deleted.

Note that when you start the system, the contents of the */usr/adm/pacct* file is usually saved in the file */usr/adm/opacct*.

4.6.2 Displaying Accounting Information

The **acctcom** command reads processing accounting information from the */usr/adm/pacct* file by default, then displays selected information on your terminal screen. The command usually displays basic accounting information, such as the process's program name, the name of the user who invoked the process, the start and stop times for the process, and the number of execution seconds in real time and CPU time. The command has several options that can be used to display selected information.

To display the average size of each process, enter:

```
acctcom
```

The command displays the basic information plus the average size of each process.

To display basic accounting information about a specific command, enter:

```
acctcom -n command
```

where *command* is the name of the command you are interested in. The command responds by displaying each entry for the specified *command*. For example, to display each entry for the system command, enter:

```
acctcom -n units
```

displays each entry for the system command **units**.

To display information about the number and size of input and output counts, enter:

```
acctcom -i
```

The command displays basic program information plus the number of characters and blocks transferred or read by each program.

To display information about a program's use of system resources, enter:

```
acctcom -h
```

The command displays the basic information plus the "use factor." The use factor is a number generated and used by the system to determine how each process should be scheduled for execution. Processes with high use factors use a high percentage of the system resources and are therefore scheduled after processes with lower factors.

Chapter 5
Maintaining File Systems

5.1 Introduction

File system maintenance, an important task of the system manager, keeps the XENIX system running smoothly, keeps the file systems clean, and ensures adequate space for all users. To maintain the file systems, the system manager must monitor the free space in each file system, and take corrective action whenever it gets too low.

This chapter explains the file system maintenance commands. These commands report how much space is used, locate seldom-used files, and remove or repair damaged files.

5.2 Maintaining Free Space

The XENIX system operates best when at least 15% of the space in each file system is free. In any system, the amount of free space depends on the size of the disk containing the file system and the number of files on the disk. Since all disks have a fixed amount of space, it is important to carefully control the number of files stored on the disk.

If a file system has less than 15% free space, system operation usually becomes sluggish. If no free space is available, the system stops any attempts to write to the file system. This means that the user's normal work on the computer (creating new files and expanding existing ones) stops.

The only remedy for a file system which has less than 15% free space is to delete one or more files from the file system. The following sections describe strategies for keeping the free space available.

5.2.1 Strategies for Maintaining Free Space

The system manager should regularly check the amount of free space of all mounted file systems and remind users to keep their directories free of unused files. You can remind users by including a reminder in the message of the day file */etc/motd*. (See the section, "Changing the */etc/motd* File" in Chapter 8).

If the amount of free space slips below 15%, the system manager should:

1. Send a system-wide message asking users to remove unused files.
2. Locate exceptionally large directories and files, and send mail to the owner asking him to remove unnecessary files.
3. Locate and remove temporary files and files named *core*.
4. Clear the contents of system log files.

Finally, if the system is chronically short of free space, it may be necessary to create and mount an additional file system.

5.2.2 Displaying Free Space

You can find out how much free space exists in a particular file system with the **df** (for "disk free") command. This command displays the number of "blocks" available on the specific file system. A block is 512 characters (or bytes) of data.

The **df** command has the form:

```
df specialfile
```

where *specialfile* can be the name of a XENIX special file corresponding to the disk drive containing the file system (see Appendix A, "XENIX Special Device Files"). If you do not give a special filename, then the free space of all normally mounted file systems is given.

For example, to display the free space of the root file system */dev/root*, enter:

```
df /dev/root
```

and press the RETURN key. The command displays the special filename and the number of free blocks. You may compute the percentage of free space by comparing the displayed value with the total number of blocks in the file system. See Appendix A, "XENIX Special Device Files," for a list of the total blocks.

5.2.3 Sending a System-Wide Message

If free space is low, you may send a message to all users on the system with the **wall** (for "write to all") command. This command copies the messages you enter at your terminal to the terminals of all users currently logged in.

To send a message, enter:

```
wall
```

and press the RETURN key. Enter the message, pressing the RETURN key to start a new line if necessary. After you have entered the message, press Ctrl-D. The command displays the message on all terminals in the system. To leave the **wall** command, press Ctrl-D. This removes the link to other terminals.

5.2.4 Displaying Disk Usage

You can display the number of blocks used within a directory by using the **du** command. This command is useful for finding excessively large directories and files.

The **du** command has the form:

```
du directory
```

The optional *directory* must be the name of a directory in a mounted file system. If you do not give a directory name, the command displays the number of blocks in the current directory.

For example, to display the number of blocks used in the directory */usr/johnd*, enter:

```
du /usr/johnd
```

and press the RETURN key. The command displays the name of each file and directory in the */usr/johnd* directory and the number of blocks used.

5.2.5 Displaying Blocks by Owner

You can display a list of users and the number of blocks they own by using the **quot** (for "quota") command. The command has the form:

```
quot specialfile
```

The *specialfile* must be the name of the special file corresponding to the disk drive containing the file system (see Appendix A, "XENIX Special Devices Files").

For example, to display the owners of files in the file system mounted on the disk drive */dev/fd1*, enter:

```
quot /dev/fd1
```

and press the RETURN key. The command displays the users who have files in the file system and the number of blocks in these files.

5.2.6 Mailing a Message to a User

If a particular user has excessively large directories or files, you may send a personal message to the user with the **mail** command.

To begin sending a message through the mail, enter:

mail *login-name*

and press the RETURN key. The *login-name* must be the login name of the recipient. To send a message, enter the message, press the RETURN key, and then press Ctrl-D. If the message has more than one line, press the RETURN key at the end of each line. The **mail** command copies the message to the user's mailbox, where he may view it also by using the **mail** command. See the XENIX *User's Guide* for details.

5.2.7 Locating Files

You may locate all files with a specified name, size, date, owner, and/or last access date by using the **find** command. The command is useful for locating seldom-used and excessively large files.

The **find** command has the form:

find *directory parameters*

The *directory* must be the name of the first directory to be searched. (It will also search all directories within that directory.) The parameters are special names and values that tell the command what to search for (see *find*(C) in the XENIX *Reference Manual* for complete details). The most useful *parameters* are:

-name *file*

-atime *number*

-print

The "-name" parameter causes the command to look for the specified *file*. The "-atime" parameter causes the command to search for files which have not been accessed for the *number* of days. The "-print" parameter causes the command to display the locations of any files it finds.

For example, to locate all files named *core* in the directory */usr*, enter:

```
find /usr -name core -print
```

and press the RETURN key. The command displays the locations of all files it finds.

5.2.8 Locating *core* and Temporary Files

You can locate *core*, and temporary files with the **find** command.

A *core* file contains a copy of a terminated program. The XENIX system sometimes creates such a file when a program causes an error from which it cannot recover. A temporary file contains data created as an intermediate step during execution of a program. These files may be left behind if a program contained an error or was prematurely stopped by the user. The name of a temporary file depends on the program that created it.

In most cases, the user has no use for either *core* or temporary files, and they can be safely removed.

When searching for *core* or temporary files, it is a good idea to search for files which have not been accessed for a reasonable period of time. For example, to find all *core* files in the */usr* directory which have not been accessed for a week, enter:

```
find /usr -name core -atime +7 -print
```

and press the RETURN key.

5.2.9 Clearing Log Files

The XENIX system maintains a number of files, called log files, that contain information about system usage. When new information is generated, the system automatically appends this information to the end of the corresponding file, preserving the file's previous contents. This means the size of each file grows as new information is appended. Since the log files can rapidly become quite large, it is important to periodically clear the files by deleting their contents.

You can clear a log file by entering:

```
cat < /dev/null > filename
```

where *filename* is the full pathname of the log file you wish to clear. A log file normally receives information to be used by one and only one program, so its name usually refers to that program. Similarly, the format of a file depends on the program that uses it. See Appendix B, "XENIX Files and Directories," for descriptions of the log files.

In some cases, clearing a file affects the subsequent output of the corresponding program. For example, clearing the file */etc/ddate* forces the next backup to be a periodic backup (see Chapter 6, "Backing Up File Systems").

5.2.10 Expanding the File System

If free space is chronically low, it may be to your advantage to expand the system's storage capacity by creating and mounting a new file system. Once mounted, you may use this new file system for your work, or even copy user or system directories to it.

A chronic shortage of space usually results from having more users on the system than the current hard disk can reasonably handle, or having too many directories or files. In either case, creating a new file system allows some of the users and directories to be transferred from the hard disk, freeing a significant amount of space on the existing file system and improving system operation. For details about creating and mounting file systems, see Chapter 4, "Using File Systems."

5.3 File System Integrity

Since file systems are normally stored on hard and floppy disks, occasional loss of data from the file system through accidental damage to the disks is not unusual. Such damage can be caused by conditions such as an improper system shutdown, hardware errors in the disk drives, or a worn out disk.

Such damage usually affects one or two files, making them inaccessible. In very rare cases, the damage causes the entire file system to become inaccessible.

The XENIX system provides a way to restore and repair a file system if it has been damaged. The **fsck** (for "file system check") command checks the consistency of file systems and, if necessary, repairs them. The command does its best to restore the information required to access the files, but it cannot restore the contents of a file once they are lost. The only way to restore lost data is to use backup files. For details about backup disks, see Chapter 6, "Backing Up File Systems."

5.3.1 Repairing the File System

You can repair a file system with the **fsck** command. The command has the form:

fsck *specialfile*

The *specialfile* must be the name of the special file corresponding to the disk drive containing the file system (see Appendix A, "XENIX Special Device Files").

For example, to check the file system on the disk in the disk drive */dev/fd1*, enter:

```
fsck /dev/fd1
```

and press the RETURN key. The program checks the file system and reports on its progress with the following messages.

```
** Phase 1 - Check Blocks and Sizes
** Phase 2 - Pathnames
** Phase 3 - Connectivity
** Phase 4 - Reference Counts
** Phase 5 - Check Free List
```

If a damaged file is found during any one of these phases, the command asks if it should be repaired or salvaged. Enter *y* to repair a damaged file. You should always allow the system to repair damaged files even if you have copies of the files elsewhere or intend to delete the damaged files.

Note that the **fsck** command deletes any file that it considers too damaged to be repaired. If you suspect a file system problem and wish to try to save some of the damaged file or files, check other possible remedies before you invoke the command.

5.3.2 Automatic File System Check

The XENIX system sometimes requests a check of the file system when you first start it. This usually occurs after an improper shutdown (for example, after a power loss). The file system check repairs any files disrupted during the shutdown. For details, see the section "Cleaning the File System" in Chapter 2.

Chapter 6
Backing Up File Systems

6.1 Introduction

A file system backup is a copy, on floppy disk, of the files in the root directory and other regularly mounted file systems. A backup allows the system manager to save a copy of the file system as it was at a specific time. The copy may be used later to restore files that are accidentally lost or temporarily removed from the file system to save space.

This chapter explains how to create backups of the root directory and other file systems, and how to restore files from the backups.

6.2 Strategies for Backups

The system manager should back up the root directory (and any other mounted file systems) on a regular basis. In particular, the manager should make daily copies of all files modified during the day, and should make periodic (e.g., weekly) copies of the entire root directory and other mounted file systems.

The XENIX system offers two ways to back up file systems, the **sysadmin** program and the **tar** command.

The **sysadmin** program is a formal maintenance program for systems that require a rigorous schedule of file system backups. The program automatically locates modified files, copies them to backup media, and optionally produces a list of the files. If your system has many users and a large number of files that are modified daily, use **sysadmin** to make regular backups.

The **tar** command is useful on systems with one or two users, or on any system where ordinary users wish to make personal copies of their directories and files. The command lets the system manager or user choose the files and directories to be copied. The command does not, however, automatically locate modified files.

A typical backup schedule includes a daily backup once a day and a periodic backup once a week. A daily backup copies only those files modified during that day; a periodic backup copies all files in the file system. The appropriate schedule for a system depends on how heavily the system is used, and how often files are modified. In all cases, a periodic backup should be done at least once a month.

The system manager should schedule backups at times when few (if any) users are on the system. This ensures that the most recent version of each file is copied correctly.

A regular schedule of backups requires a large number of floppy disks and adequate storage for the disks. Daily backups should be saved at least two weeks; periodic backups should be saved indefinitely. Disks should be

properly labeled with the date of the backup and the names of the files and directories contained in the backup. After a backup has expired, the disk may be used to create new backups.

Note

If the number of floppies needed for making backups grows too large, the system manager can use the **backup** command instead of **sysadmin**. **backup** is called by **sysadmin** and by using **backup** directly. The system administrator can tailor the number of floppies used to fit the needs of the individual site. Refer to **backup**(C) in the XENIX *Reference*.

6.3 Using the sysadmin Program

The **sysadmin** program performs daily and periodic backups, lists backup files, and restores individual files from backup disks. The program presents each task as an item in a menu. To perform a task, simply choose the appropriate item from the menu and supply the required information.

6.3.1 Creating Backups

To create backups with the **sysadmin** program, you need several formatted floppy disks. The exact number depends on the number of files to be copied; for example, some periodic backups require as many as nine disks. For details on how to format a floppy disk, see the section "Formatting Floppy Disks" in Chapter 4.

To create a backup, follow these steps:

1. Log in as the super-user.
2. Enter:

   sysadmin

   and press the RETURN key. The program displays a file system maintenance menu.

```
                File System Maintenance
                ------------------------------------
Type    1 to do daily backup
        2 to do a periodic backup
        3 to get a backup listing
        4 to restore a file
        5 to quit
```

3. Enter *1* for a daily backup or *2* for a periodic backup. Then press the RETURN key. Note that if the system has never had a periodic backup, it automatically performs one, even if you have chosen a daily backup.

4. Insert a floppy disk in drive 1, wait for the drive to accept the disk (all drive noise should stop), and press the RETURN key. The system displays the current date and the date of the last backup (it displays "the epoch" if there has been no backup). The system then begins to copy files to the floppy disks. If the disk runs out of space, the program displays the message:

```
Change volumes
```

5. Remove the first disk and insert a new disk. Wait for the drive to accept the disk, then press the RETURN key. The program continues to copy files to the new disk. Repeat this step until the program displays the message:

```
DONE
```

When doing a periodic backup, you may need to repeat the last step several times before the backup is complete. You should label each disk as you remove it from the disk drive. For example, label the first disk "Volume 1", the second "Volume 2", and so on.

6.3.2 Getting a Backup Listing

You can keep a record of the files you have backed up by invoking the **sysadmin** program and selecting the third item in the menu. The program copies the names of all files from the backup disks to the temporary file */tmp/backup.list*. This listing is especially convenient if you keep detailed records of the files copied in each backup. The backup listing is available after every daily or periodic backup.

To get the listing, follow these steps:

1. Log in as the super-user.
2. Enter:

sysadmin

and press the RETURN key. The program displays the system maintenance menu.

3. Enter *3* and press the RETURN key. The program prompts you to reinsert the backup disks in the same order that you inserted them during the backup.

4. Insert the first disk, wait until the drive accepts the disk, then press the RETURN key. The program automatically reads the filenames off the backup disk and places them in the list file. When the program has read all the names, it asks for the next disk.

5. Remove the first disk and insert the next. Wait for the drive to accept the disk and press the RETURN key. Repeat this step until all disks have been read.

You may produce a printed copy of the backup list by printing the list at the lineprinter. Enter:

lpr /tmp/backup.list

and press the RETURN key. To save space after printing the file, you should remove it from the */tmp* directory with the **rm** command.

6.3.3 Restoring a Backup File

You can restore files from the backup disks by invoking the **sysadmin** program and selecting the fourth item in the menu. You will need the complete set of backup disks containing the latest version of the file you wish to restore. You will also need the "full pathname" of the file you wish to restore. This is the name given for the file in the backup listing.

To restore a file, follow these steps:

1. Log in as the super-user.

2. Enter:

sysadmin

and press the RETURN key. The program displays the file system maintenance menu.

3. Enter *4* and press the RETURN key. The program prompts you to enter the full pathname of the file you wish to restore.

4. Enter the pathname and press the RETURN key. The program prompts for another pathname.

5. Repeat step 4 to enter another pathname, or press the RETURN key to continue the program. If you press the RETURN key, the program prompts you to insert the first disk in the backup set.

6. Insert the first disk in the set of backup disks (volume 1), wait for the drive to accept the disk, and press the RETURN key. The program displays the inode numbers of the files you have given, then prompts for the volume number of the backup disk containing the files.

7. Insert the disk having the correct volume number, enter the volume number, and press the RETURN key. The program searches the disk for the specified files. If found, the files are copied to your current directory. If not found, the program prompts for the next volume.

8. Repeat step 7 until all files have been found and copied.

The **sysadmin** program does not restore the file's original name. Instead, it names the file a unique number called an "inode" number. You can restore the file's original name using the **mv** (for "move") command:

mv *inode filename*

inode is the name given to the file by **sysadmin**. *filenme* is the new name you want for the file. For example, to restore a file */usr/johnd/projects.june* from *224*, enter:

```
mv  224  /usr/johnd/projects.june
```

6.4 Using the tar Command

The **tar** command copies specified files and directories to and from floppy disks. On systems with one or more users, it gives the system manager a direct way to make backup copies of the files modified during a day. On systems with many users, it gives ordinary users a way to make personal copies of their own files and directories.

6.4.1 Copying Files to a tar Disk

You can copy a small number of files or directories to a floppy disk with the **tar** command. The command has the form:

tar cvf *specialfile files*

The *specialfile* must be the name of the special file corresponding to a disk drive (see Appendix A, "XENIX Special Device Files"). The drive must contain a formatted disk. The *files* are the names of the files or directories you wish to copy.

To use the **tar** command, you need a formatted floppy disk and the names of the files and/or directories you wish to copy. For details about how to format a disk, see the section "Formatting Floppy Disks" in Chapter 4. If you give a directory name, the command copies all files in the directory (including subdirectories) to the disk.

For example, to copy the files a, b, and c to the disk in the disk drive */dev/fd1*, enter:

```
tar cvf /dev/fd1 a b c
```

and press the RETURN key.

6.4.2 Restoring Files From a tar Disk

You may also use the **tar** command to restore files from a disk. The command simply copies all files on the disk to your current directory. In this case, the command has the form:

```
tar xvf specialfile
```

The *specialfile* must be the name of the special file corresponding to the disk drive containing the **tar** disk.

For example, to restore files from the disk in the drive */dev/fd1*, enter:

```
tar xvf /dev/fd1
```

and press the RETURN key. The command copies files on the disk in the drive to the current directory.

Since the **tar** command copies files only to the current directory, make sure you are in the desired directory before you invoke the command. You can change to the desired directory with the **cd** command.

Chapter 7

Using Peripheral Devices

7.1 Introduction

One important task of the system manager is to add peripheral devices such as terminals, hard disks, and lineprinters to the system. Adding these devices lets more users access the system, gives extra storage space for user's files and directories, and adds to the system's overall capabilities.

To add a peripheral device, the system manager must make the physical connection between the device and the computer, then use the correct system commands to enable the device for operation. This chapter explains how to do this and how to maintain the devices once they are added.

Note that all physical connections between a device and the system are device-dependent. For information about these connections, see the hardware manual provided with the device and your computer.

7.2 Using Multiscreen

The multiple screen feature uses the console and */dev/tty[02...10]* device files. These device files provide character I/O between the system and the computer's screen and keyboard. The actual number of multiscreens available is a function of the amount of memory in your personal computer and is displayed when the system boots. There are generally between two and six multiscreens enabled.

Each device file has an independent screen display and keyboard access. The files are used chiefly to display characters at the screen devices and read characters from the keyboard.

Although the multiscreens may be open and active at the same time, only one can be seen at any given time. The selected multiscreen is the terminal currently "connected" to the keyboard. Rotate through the screens by pressing **Ctrl-PrtSc** (using the **Ctrl** key and the **PrtSc** key, the shifted asterisk, directly below carriage return). Any active screen may be selected by pressing **alt-F***n where* **F***n* is one of the ten function keys on the far left side of the keyboard. For example, entering:

```
alt-F6
```

will switch you to screen 6, corresponding to */dev/tty06*. Refer to the XENIX *Reference* for more on **multiscreen**(M) and **console**(M).

Note that error messages from the kernel appear on the console screen. In such cases, the console screen displays instead of the current screen.

The system displays the number of multiscreens available on boot. This number is determined by the amount of memory in your computer. Do not try to access more than the auto-configured number of screens.

7.3 Adding a Terminal

You can give two or more users simultaneous access to the system by adding extra terminals. You can add a terminal by connecting it to an RS-232 serial line on the system and enabling it with the **enable**(C) command. Many different terminals work well with the XENIX operating system. A short list of recommended models is given in **terminals**(M) in the XENIX *Reference Manual.*

Before you add a terminal, you must know how to connect the terminal to a serial line on the computer. You also need to know the name of the serial line. Physical connections for the terminal are usually explained in the terminal's hardware manual. The names of the system's available serial lines are given in Appendix A, "XENIX Special Device Files." Once a terminal has been connected, you may then enable the terminal for use with the **enable** command.

To add a terminal, follow these steps:

1. Using the recommended procedure in the terminal's hardware manual, connect the terminal to one of the computer's RS-232 serial lines. Make sure that the terminal is compatible with the line configuration.

2. Log in as the super-user.

3. Use the **enable** command to enable the terminal. The command has the form:

   enable *specialfile*

   where *specialfile* is the name of the serial line to which the terminal is attached. This name depends on your system's configuration (see Appendix A, "XENIX Special Device Files"). For example, the command

   enable /dev/tty11

   enables the terminal connected on serial line */dev/tty11*.

4. Turn on the power to the terminal and press the RETURN key several times. The system should display a "login:" message. When it does, you may log in and begin work.

If no "login:" message appears on the screen, if random characters appear, or if the terminal does not respond to your attempt to log in, you may need to change the baud rate (or line speed) of the terminal to match the serial line. You can change the baud rate with the **stty**(C) command described in the next section.

When using the **enable** command, make sure that you wait a full minute between each use of the command. Failure to do so can cause a system crash.

7.4 Setting Terminal Lines

Your XENIX system can adapt itself to several different terminal baud rates and settings. The same program that displays the login message, **getty**(C), reads these terminal line values from a table, trying each setting until one is successful, and the user can log in to the system. This table provides several default settings for different kinds of terminals lines.

On your XENIX system, **getty** automatically executes as part of the login process. The table of terminal settings is found in a file called */etc/gettydefs*. You can edit *gettydefs* to add different sets of terminal characteristics or to change the existing ones.

7.4.1 The gettydefs File

The file */etc/gettydefs* contains the information that **getty** uses to set up terminal line characteristics such as baud rate. The file is in the form of a table. Each table entry is divided into five fields. These fields include:

label# initial-flags # final-flags #
login-prompt #next-label[*#login program*]

where:

label — Identifies the *gettydefs* entry to **getty**. This could be a number or a letter. *Label* corresponds to the line mode field in */etc/ttys*. **Init** passes the line mode to **getty** as an argument.

initial-flags — Sets terminal line characteristics when **getty** first establishes the connection with the line. **getty** recognizes the flags listed in *tty*, XENIX *Reference Manual*(M). Often the only flag to appear in this field is the one setting the baud rate. For example, B300 would set the speed to 300.

final-flags — Sets the terminal line characteristics just before **getty** executes **login**. These flags describe the operating characteristics for the line. The baud rate (B) is set again. Other common flags include SANE (a composite flag that sets a number of terminal characteristics to reasonable values), TAB3 (expands tabs with spaces), IXANY (enables any character to restart

output), and HUPCL (hangs up line on final close). Flags can be entered in any order.

login-prompt Contains the login prompt message that greets users. This field is printed exactly as it is entered including spaces and tabs. The '@' in the login-prompt field is expanded to the first line in the file */etc/systemid* (unless the '@' is preceded by a '\').

Several character sequences are recognized, including:

\n	Line feed	\t	Tab
\r	Carriage return	\f	Form feed
\v	Vertical tab	\b	Backspace
nnn	(3 octal digits) The specified ASCII characters		

next-label Identifies the next label in *gettydefs* for **getty** to try if the current one is not successful. **getty** tries the next label if a user presses the BREAK key while attempting to log in to the system. Groups of entries, such as dial-up or TTY lines, should form a closed set so that **getty** cycles back to the original entry if none of the entries is successful.

login-program The name of the program which actually logs users onto XENIX. The default program is */etc/login*.

If preceeded by the keyword "AUTO," **getty** does not prompt for a username, but instead uses its first argument as the username and immediately executes the *login-program*

Each field is separated by pound sign (#), and each entry in *gettydefs* is separated by a blank line.

An entry in *gettydefs* might look like this:

```
4# B1200 # B1200 SANE TAB3 HUPCL #Login: #2#AUTO
```

The number 4 identifies this entry to **getty**. In the next field, the baud rate is set to 1200. The final settings in the third field include the baud rate (B1200), SANE (a composite flag for a number of characteristics), and HUPCL (hangs up line on final close). The login prompt appears as "Login:", and if this setting is not successful, **getty** proceeds to label 2 in

gettydefs. AUTO attempts to log in the user "*/dev/console* (or other tty)" executing */etc/login*.

If the last entry also contains a file name, that login program is executed. For example, including a file such as */etc/dial_login* for a line connected to a modem can be used. It would set the user ID, acquire a password, validate the user, and then become the user. It could possibly require a password for the system in addition to an account password and even have a special set of login environment variables included in */etc/default/dial_login*.

7.4.2 Changing the gettydefs File

The file */etc/gettydefs* already exists on your XENIX system and has sets of entries for the operator's console, dial-up lines, and terminal lines. These different sets correspond to line mode settings in */etc/ttys*. The **init** program passes the line mode as an argument to **getty**.

You can edit *gettydefs* to add new terminal settings or to change existing ones. For example, the settings for terminal lines on your XENIX system might look like this:

```
1# B2400 # B2400 SANE IXANY TAB3 #@!login: #2

2# B4800 #B4800 SANE IXANY TAB3 #@!login: #3

3# B9600 # B9600 SANE IXANY TAB3 #@!login: #1
```

To change the sample *gettydefs* file so that the first baud rate **getty** attempts is 1200, do the following:

1. Enter a text editor to edit the first line of the file *gettydefs*.
2. Change the second and third fields from B2400 to B1200.
3. Exit the text editor, saving *gettydefs*.

The sample file should look like this:

```
1# B1200 # B1200 SANE IXANY TAB3 #@!login: #2
2# B4800 #B4800 SANE IXANY TAB3 #@!login: #3
3# B9600 # B9600 SANE IXANY TAB3 #@!login: #1
```

You can also add additional terminal line settings to *gettydefs*. Flags and permissible values for terminal settings are listed in **tty** (M), XENIX *Reference Manual*.

When you add a new entry, be sure that the groups of entries in *gettydefs* form a closed set, so the *next-label* field of the last entry directs **getty** back to the first entry in the group.

To add an entry for a baud rate of 300 to the preceding sample *gettydefs* file, follow these steps:

1. Enter a text editor to edit the file */etc/gettydefs*.

2. Locate the point where you want to insert the new settings for *gettydefs*. The order of the entries does not matter; **getty** only looks for the label. In this example, the new entry will be the last entry in the file.

3. Insert a carriage return after the last line in the file and enter the following on a new line:

   ```
   4# B300 # B300 SANE IXANY TAB3 #@!login: #1
   ```

4. To incorporate label 4 into the set of labels, change the *next label* field for entry 3 to 4:

   ```
   3# B4800 # B4800 SANE IXANY TAB3 #@!login: #4
   ```

   getty is now directed from label 3 to 4, and then back to 1.

5. Exit the text editor, saving the revised *gettydefs* file.

The new *gettydefs* looks like this:

```
1# B1200 # B1200 SANE IXANY TAB3 #@!login: #2
2# B4800 #B4800 SANE IXANY TAB3 #@!login: #3
3# B9600 # B9600 SANE IXANY TAB3 #@!login: #4
4# B300 # B300 SANE IXANY TAB3 #@!login: #1
```

7.4.3 Checking the Terminal Settings

Each time you change the terminal line settings or add new entries to *gettydefs*, you should check to make sure that the new values that make sense to **getty**. To do this you use the command **getty** with the check option, **-c**, and the filename.

For example, to check *gettydefs*, enter:

```
getty -c /etc/gettydefs
```

If any of the values and settings in *gettydefs* are not permitted, **getty -c** displays them on your terminal screen.

For more information on **getty** and *gettydefs*, see **getty**(M) and **gettydefs**(F), XENIX *Reference* Manual.

7.5 Changing Serial Line Operation

Whenever you enable a terminal with the **enable** command, the system automatically sets the operating characteristics of the serial line to a set of default values. Sometimes these values do not match the values used by the terminal and, therefore, must be changed to allow communication between the system and the terminal. You can display and change the operating characteristics of a serial line with the **stty** (for "set tty") command.

You can display the current operating characteristics of a serial line by entering:

```
stty
```

at the terminal connected to that line. If it is impossible to log in at that terminal, you may use another terminal to display the characteristics. Log in as the super-user at another terminal, and enter:

```
stty <specialfile
```

where *specialfile* is the name of the device special file corresponding to the serial line (see Appendix A, "XENIX Special Device Files"). For example, the command:

```
stty </dev/tty01
```

displays the current characteristics of the serial line named */dev/tty01*. The command displays the baud rate, the parity scheme, and other information about the serial line. The meaning of this information is explained in **stty**(C) in the XENIX *Reference Manual*.

One common change to a serial line is changing the baud rate. This is usually done from a terminal connected to another serial line since changing the rate disrupts communication between the terminal and system. Before you can change the rate, you need to know the current baud rate of the terminal (review the terminal's hardware manual to see how to determine the terminal's current rate). Once you have the baud rate, log in as the super-user at the other terminal, and enter:

```
stty baud-rate  <specialfile
```

where *baud-rate* is the terminal's current baud rate, and *special* is the name of the device special file corresponding to the serial line you wish to change. The baud rate must be in the set 50, 75, 110, 134, 150, 200, 300, 600, 1200, 2400, 4800, and 9600. For example, the command:

```
stty 9600 </dev/tty01
```

changes the baud rate of the serial line */dev/tty01* to 9600. Note that the "less than" symbol (<) is used for both displaying and setting the serial line from another terminal.

Another common change is changing the way the system processes input and output through the serial line. Such changes are usually made from the terminal connected to the serial line. For example, the command:

```
stty tabs
```

causes the system to expand tabs with spaces (used with terminals which do not expand tabs on their own), and the command:

```
stty echoe
```

causes the system to remove a deleted character from the terminal screen when you back over it with the BACKSPACE key.

Note that the **stty** command may also be used to adapt a serial line to an unusual terminal, to another type of serial device which requires parity generation and detection, or special input and output processing.

For a full description of this command, see **stty**(C) in the XENIX *Reference Manual*.

7.6 Setting the Terminal Type

The XENIX system requires that an enabled terminal's type be clearly defined before any work is done at the terminal. You can set the terminal type by assigning the type to the TERM variable, a special XENIX system variable that associates the terminal you are using with a list of characteristics given in the */etc/termcap* file. The characteristics tell the system how to interpret your terminal's keys and how to display data on your terminal's screen.

The TERM assignment has the form:

```
TERM=termtype ; export TERM
```

The *termtype* must be one of the names associated with one of the terminals defined in the */etc/termcap* file. The assignment must be entered at the terminal whose type you are setting.

For example, to set the terminal type to "ansi", go to the terminal you wish to set, enter:

```
TERM=ansi ; export TERM
```

and press the RETURN key.

If you are not sure which name you may use for *termtype*, you can view the names by displaying the */etc/termcap* file. To display the file, enter:

```
cat /etc/termcap
```

and press the RETURN key. Since the file is very large, you will need to use the Ctrl-S key to stop the display at every full screen. You may view more of the file by pressing the Ctrl-Q key or the Spacebar.

You can let the system define the terminal type automatically whenever you log in by including the TERM assignment in your *.profile* file (see Chapter 8 "Solving System Problems" under "Changing the *.profile* File").

If you do let the system set the terminal type be careful when logging in on terminals that are not the same as your normal terminal. The XENIX system has no way of checking whether or not the terminal assignment is correct for the given terminal and assumes that it is the same as your normal terminal. If it is not, you must set the terminal type manually.

7.7 Removing a Terminal

From time to time it may be necessary to remove a terminal from the system, for example, if you wish to replace it with some other device. Before you can remove a terminal, you must disable it with the **disable**(C) command.

To remove a terminal, follow these steps:

1. Turn off the power to the terminal.

2. Log in as the super-user at another terminal.

3. Use the **disable** command to disable the terminal. The command has the form:

   disable *specialfile*

   where *specialfile* is the name of the serial line to which the terminal is attached. For example, the command:

   disable /dev/tty01

   disables the terminal connected to serial line */dev/tty01*.

4. Disconnect the terminal from the system.

The serial line previously connected to the terminal is now free to accept another device.

When using the **disable** command, make sure that you wait a full minute between each use of the command. Failure to do so can cause a system crash.

7.8 Modem Usage under XENIX

7.8.1 Serial Lines

XENIX supports modem control on up to two serial ports. The following device names refer to the serial ports with and without modem control.

Device:	Function:
/dev/tty11	main serial adaptor **without** modem control.
/dev/tty13	main serial adaptor **with** modem control.
/dev/tty12	alternate serial adaptor **without** modem control.
/dev/tty14	alternate serial adaptor **with** modem control.

/dev/tty11 and */dev/tty13* refer to the same serial port (likewise for */dev/tty12* and */dev/tty14*). The operating system uses different device-driver subroutines for each. Never attempt to use both modem and non-modem control ports at the same time or you will see the warning:

"cannot open: device busy"

7.8.2 Dialing Out From Your Computer

The **cu**(C) and **uucp**(C) utilities are used to call remote systems and transfer data under XENIX. The file */usr/lib/uucp/L-devices* (refered to as *L-devices*) contains information used by these programs to determine the characteristics of a particular serial. The *L-devices* file comes set for the most common uses of **cu** and **uucp**.

The *L-devices* file contains lines which specify the device for the line, the call-unit associated with the line (0 for direct lines), and the baud rate, which are to be used by **uucp**. The following table shows example *L-devices* lines for various types of connections (direct or modem):

For Outgoing Calls					
Serial Line	**Connection**	***L-devices* line**			
main serial adapter	(direct)	DIR	tty11	0	1200*
main serial adapter	(modem)	ACU	tty11	tty11	1200*
alternate serial adapter	(direct)	DIR	tty12	0	9600*
alternate serial adapter	(modem)	ACU	tty12	tty12	300*
* example baud rate. Substitute the baud rate for your machine or modem.					
Direct hookups depend on the baud rate setting of the login on the remote machine.					
Modem hookups depend on the capability of your modem and the modem you are dialing.					

Modem control lines are only used on lines dialing in. For outgoing calls, use lines with no modem control.

Note that these *L-devices* lines are compatible alternatives; they can exist in your *L-devices* file at the same time.

A dial out line must be disabled. To disable a line, enter:

```
disable tty
```

This command disables */dev/tty11* for dialing out:

```
disable /dev/tty11
```

To invoke **cu** for a direct line, enter:

```
cu dir
```

cu will select the first device from the *L-devices* file that matches the connection type, (DIR or ACU) and the speed. The default speed is 1200 baud. To select a specific line use the **-l** option. For example:

```
cu -l tty12 dir
```

This makes specific use of tty12. To request a speed other than the default (1200), request use the **-s** option:

```
cu -s 9600 dir
```

This will find the first direct, 9600 baud line, in this case tty12.

For dialing, both **cu** and **uucp** use the file an executable **C** program. This program dials a Hayes® Smartmodem 1200 or Smartmodem 1200B (a plug in modem card).

The source for **dial** is in the directory */usr/lib/uucp* along with the source for a program which dials a Racal Vadic 3450 modem and a makefile for use in recompiling these two source programs. The Racal Vadic source is in */usr/lib/uucp/dial_racal.c*. If you have a Racal Vadic 3450 and the XENIX Development System, you can compile *dial_racal.c* with the **make**(CP) command and replace the original **dial** with the new binary. If you have any other kind of modem, you can modify either of the two source files and create your own **dial** program.

To replace the **dial** program with another program, follow these steps:

1. Change directory to

   ```
   cd /usr/lib/uucp
   ```

2. Enter:

   ```
   mv dial dial.00
   ```

   This backs up the old **dial** program.

3. If you modified *dial_racal.c* to work with your modem, or you are using a Racal Vadic 3450 modem, enter:

   ```
   mv dial.c dial_hayes.c
   mv dial_racal.c dial.c
   ```

   Now enter the command:

   ```
   make
   ```

   and press RETURN

4. If you modified the file *dial.c*, or are just recompiling the **dial** program, do not follow the previous step. Enter:

   ```
   make
   ```

   and press RETURN.

5. When the **make** is finished, you will have a new */usr/lib/uucp/dial* program.

The dialer program can also be a **sh**(C) script.

When you are hooking up your modem, or any other device, make sure that serial wires connected to your computer are not left hanging. An unterminated line connected to your computer can considerably reduce system performance. Unplug a modem wire at the computer.

7.8.3 Dialing Into Your Computer Under XENIX

To allow dialing into your computer, you must enable a serial line that recognizes modem control signals, with the **enable**(C) command. When using the **enable** command, make sure that you wait a full minute between each use of the command. Failure to do so may send too many signals to the **init**(C) program, which will then terminate. If **init** terminates, no new logins are possible.

To use the main serial adapter, enter:

```
disable tty11
enable tty13
```

Or, for the the alternate serial adapter, enter:

```
disable tty12
enable tty14
```

Note that tty13 and tty11 refer to the same (main) serial line, and tty14 and tty12 refer to the same (alternate) serial line. Do not use the same line in both its modem and non-modem invocations at the same time as this will cause an error.

7.8.4 Hayes Modem Settings

Proper modem configuration is necessary when using **cu** and **uucp**. Modem settings differ for each modem. Consult your modem manual for the proper switch settings.

If you have a Hayes Smartmodem 1200, switches 3 and 8 should be down:

	1	2	3	4	5	6	7	8
up	●	●		●	●	●	●	
down			●					●

This allows both the XENIX and DOS communications systems to work.

/usr/lib/uucp/L.sys is the file containing information about systems to which **uucp** can connect. In order to prefix a number in *L.sys* with a non-numeric character, that character must be defined in the file */usr/lib/uucp/L-dialcodes*. In some cases this is needed to pass special codes on to modems. For example, a modem might require the string "P" to pulse dial (to dial on a line with pulse dial, instead of touch tone dial). A sample *L-dialcodes* line is:

```
PULSE P
```

The number would be listed in *L.sys* as "PULSE4085551234". For more information on *L-dialcodes* and *L.sys*, refer to Chapter 6 in the XENIX *User's Guide* "Building a Communications System"

7.8.5 Modem Control

uucp must be used with modem control serial lines for dialing in. When dialing out, **uucp**(C) must use a no-modem serial line, and the login must be disabled on the port corresponding to that line. Check the section "Serial Lines" to see which lines support modem control.

The modem must be set to respond to DTR (Data Terminal Ready). Check your modem manual for instructions on setting DTR. Note that the dialer port must be disabled, and the port should be owned by *uucp*.

7.9 Adding a Lineprinter

7.9.1 Printer Setup

This section explains how to add new lineprinters to your XENIX system and how to use the lineprinter commands to organize and control your printing.

The XENIX lineprinter spooling system is a collection of commands that help you, as system manager, to efficiently install, monitor, and control the lineprinters serving your system. When a user requests a file to be printed using the **lp**(C) command, the lineprinter system responds with a "request ID." This consists of the name of the printer on which the file will be printed and a unique number identifying the file. With this request ID, the user can find out the status of the print request or cancel it. The **lp** options help the user to easily control printer output. For more information on **lp**, see section 4.12, "Using the Lineprinter," in the *Introduction to XENIX*.

There are several terms used to describe the line printer system:

device	The target for **lp** output. It can be a hard-wired printer, a terminal that is sometimes used as a printer, or a regular file. A device can be represented by a full XENIX pathname.
printer	The name assigned by the system manager to represent a device. This name can have up to 14 characters. At different times, a printer may be associated with different devices.
class	An ordered list of printers. Print requests sent to a class of printers are printed by the first available member of that class.
destination	A *destination* is where print requests are sent. A destination can be a class or a printer.

Consult your computer's and lineprinter's hardware manuals for information on making the connection between your system and printing devices.

7.9.2 Installing a Printer: lpinit

To install new printing devices on your XENIX system, use the **lpinit**(C) command. Before you use **lpinit** you should first know the port to which the lineprinter is connected or the XENIX pathname of the device (for example, */dev/tty03*) and the lineprinter interface program. A model interface program is supplied with your XENIX system. For more information on printer interface programs, see section 7.9.10, "Printer Interface Programs."

When you give the **lpinit** command, you will be asked a series of questions for which the default answers are displayed. If you wish to choose the default answer, simply press the RETURN key. If you want to supply your own answers, enter the information as you are prompted. If you make a mistake while responding to the questions, just press the DELETE key or the INTERRUPT key and start again.

Example

The following example shows how to add a line printer to your system. The printer name is *printer1*, the device pathname is */dev/tty30*, and it will be the default printer for your system:

1. Enter the command:

   /etc/lpinit

2. **lpinit** displays the following message on your screen:

```
If you have an unusual printer you must create an interface
program in /usr/spool/lp/model.  For a sample intrerface program
look at /usr/spool/lp/model/dumb.
```

3. Now you are prompted as to where the printer is connected. You see a menu similar to this:

```
The printer is attached to:
        1 - Main Serial Port.
        2 - Alternate Serial Port.
        3 - Parallel Port on Monochrome Card.
        4 - Main Parallel Port.
        5 - Alternate Parallel Port.

Enter one of the options above or 'q' to quit:
```

   Note that some computers do not have an Alternate Parallel Port (option 5).

   Enter the option number for you printer configuration and press RETURN.

4. **lpinit** displays the following message:

```
Enter a name for the printer or press <RETURN>
to use the default name printer:
```

   Printer names can be up to 14 characters long and can be any combination of numbers, letters, or underscore characters. Enter the printer name, *printer1*, and press the RETURN key.

5. Now you are prompted for a printer type:

```
The printer is:
        1 - dumb printer
        2 - Imagen laser printer (parallel interface)
        3 - Imagen laser printer (serial interface)
        4 - other

Enter one of the options above or type 'q' to quit:
```

Enter one of the options, then press RETURN.

6. **lpinit** displays the following message:

```
Enter the full pathname of an interface program
or press <RETURN> to use the default interface
/usr/spool/lp/model/dumb:
```

This example uses the default choice, */usr/spool/lp/model/dumb*, so simply press the RETURN key.

7. After you have finished responding to these questions, **lpinit** displays the following message:

```
scheduler stopped
destination "printer1" now accepting requests
printer "printer1" now enabled
```

This means that the lineprinter scheduling program, **lpsched**, has been momentarily stopped so that **lpinit** can add *printer1* to your system. Then, **lpinit** automatically gives the instructions to enable *printer1* and allow it to accept print requests.

8. After enabling *printer1*, **lpinit** prompts you to respond to one more question: will this printer be the system default printer?:

```
Is this the default printer ? (y/n)
```

   You can enter *Y* (for yes) or *N* (for no) depending upon whether you want users' print requests to be automatically routed to *printer1* or not. Pressing the RETURN key is the same as entering *Y*.

9. One last message is displayed:

```
If you need to modify your interface program after installing
it,  the installed version is in /usr/spool/lp/interface/dumb.
```

After you have responded to these questions, **lpinit** re-starts **lpsched**, and users can begin printing files on the new printer.

You can also add printers to your system using the **lpadmin** command discussed in the section 7.9.4, "Changing Printer Configuration." However, you will need to give separate commands to stop **lpsched**, to enable the printer, and to allow it to accept print requests. For more information on these programs and commands, see sections 7.9.3 ("Stopping the Print Spooling Daemon"), 7.9.8 ("Accepting and Rejecting Print Requests"), and 7.9.9 ("Enabling and Disabling Printers").

7.9.3 Stopping the Print Spooling Daemon: lpsched

The spooling daemon, **lpsched**, routes print requests through the correct printer interface program and then to the lineprinter. No printing can be done on your system unless **lpsched** is running. The program **lpsched** starts automatically each time your XENIX system is restarted. Sometimes it is necessary to stop **lpsched**, especially if you want to reconfigure printers or if you want to add new printers using the **lpadmin** command (**lpinit** automatically stops and restarts **lpsched** for you).

This section explains how to find out whether or not **lpsched** is running, how to stop and re-start it, and how to re-create it if necessary.

To find out whether or not **lpsched** is running, enter:

```
lpstat -r
```

In response to the **lpstat** command, your screen displays a message that **lpsched** is either running or not running.

To shutdown the scheduler, **lpsched**, enter:

```
/usr/lib/lpshut
```

lpsched stops running and all printing stops as well. Printing requests stopped in the middle of printing reprint when **lpsched** starts again.

After you have finished configuring the printers, you should re-start **lpsched**. To do this, enter:

```
/usr/lib/lpsched
```

Next, check **lpstat -r** to confirm that **lpsched** is running. If it is not, enter the following on two lines:

```
rm -f /usr/spool/lp/SCHEDLOCK
/usr/lib/lpsched
```

This removes any previous version of the file *SCHEDLOCK*. *SCHEDLOCK* assures that only one version of **lpsched** runs at a time.

Each time **lpsched** sends a print request to an interface program, it records an entry in a log file, */usr/spool/lp/log*. The entry includes the user name, the request ID, the name of the printer the request will be printed on, and the date and time requested. **lpsched** also records any error messages in this file. After you have stopped **lpsched**, the log file is renamed */usr/spool/lp/oldlog* and **lpsched** starts a new log file. Requests waiting to be printed before **lpsched** was stopped may have an entry in both log files.

For more information on **lpsched**, see *lpsched*(C), XENIX *Reference Manual*.

7.9.4 Changing Printer Configuration: lpadmin

After you have added printers to your system using **lpinit**, you may want to alter their configuration. For example, you may want to modify an interface program or create a new class of printers. To change the printer configuration, use the **lpadmin** command with the **-p** option. If **lpsched**, the printing scheduler, is running, **lpadmin** will not alter the configuration except where noted. To stop **lpsched**, see section 7.9.3 "Stopping the Print Spooling Daemon."

The **lpadmin** command has the form

/usr/lib/lpadmin -p*printer options*

where useful *options* include the following:

-c Creates a class of printers or adds a printer to a class. The same rules apply to class names as to printer names: up to 14 alphanumeric characters or underscores. Print requests sent to a class of printers will be printed on the first available member of the class.

For example, to add a printer named *laser* to a class of printers named *class1*, enter:

/usr/lib/lpadmin -plaser -cclass1

-r Removes *printer* from a class of printers. If the printer is the last member of the class, **lpadmin** removes the class as well. For example, to remove *laser* from *class1*, enter:

/usr/lib/lpadmin -plaser -rclass1

-i Assigns a new interface program to *printer*. The pathname of the interface program must follow **-i**. For example, to establish */usr/spool/lp/model/dumb* as the interface program for a printer named *prt1*, enter:

/usr/lib/lpadmin -pprt1 -i/usr/spool/lp/model/dumb

You can also use the **lpadmin** command to add new printers to your system. Unlike the **lpinit** command, **lpadmin** only configures the new printer. You must separately perform the steps of stopping **lpsched**, enabling the new printer, and giving the **accept** command so that the printer will accept print requests. For more information on using **lpadmin** to add new printers to your system, see **lpadmin**(C), in the XENIX *Reference Manual*.

7.9.5 Creating a Default Printer

You can specify one printer or one class of printers on your system to be the default destination of a print request. All files will be sent to the system default destination for printing unless the user specifies a printer when using the **lp** command. For more information on **lp** and its options, see section 4.12, *Introduction to XENIX*.

To create or change the system default destination, use **lpadmin** and option **-d**. You can change this option even when the scheduler, **lpsched**, is running.

For example, to make a class of printers called *class2* the system default destination, enter:

```
/usr/lib/lpadmin -dclass2
```

To establish a no default destination, you use the **-d** option without a printer name:

```
/usr/lib/lpadmin -d
```

For more information, see **lpadmin**(C) XENIX *Reference Manual*.

7.9.6 Removing Printers

You can remove printing destinations, either printers or classes of printers, if there are no print requests routed to them. If you remove a class of printers, the individual printers remain in the system. Removal of the last remaining printer in a class of printers removes the class as well.

To remove a printer or class of printers, use **lpadmin** and the **-x** option. For example, to remove a printer, *printer2*, from your system, enter:

```
/usr/lib/lpadmin -xprinter2
```

If you wish to temporarily remove a printer because of a breakdown, see section 7.9.9, "Enabling and Disabling Printers." If you wish to temporarily re-route print requests to other printers, see section 7.9.7,"-Moving Printing Requests Between Destinations," or section 7.9.8, "-Accepting and Rejecting Printing Requests."

For more information on **lpadmin** and its options, see **lpadmin**(C), XENIX *Reference Manual*.

7.9.7 Moving Requests Between Printers: lpmove

You can move print requests between printing destinations by using the command **lpmove**. This command does not move print requests while the scheduler, **lpsched**, is running. To stop **lpsched**, see section 7.9.3, "-Stopping the Print Spooling Daemon." **lpmove** will move individual print requests by request ID, or all requests waiting to be printed on a particular printer.

For example, to move a request with a request ID of *quick-532* to a printer named *slow*, enter:

```
/usr/lib/lpmove  quick-532  slow
```

The print request now has a new request ID: *slow-532*.

To move all requests on a printer named *quick* to *slow*, enter:

```
/usr/lib/lpmove quick slow
```

For more information on **lpmove**, see **lpsched** (C), XENIX *Reference Manual*.

7.9.8 Accepting and Rejecting Print Requests: accept

The **accept** command allows printers or classes of printers to accept print requests made with the **lp** command. You can allow a printer to accept requests after it has been properly configured. The printer, however, will not begin printing the requests until the **enable** command is given. If you added a printer to your system using the **lpinit** command, these steps were automatically performed. For information on **enable**, see section 7.9.9, "Enabling and Disabling Printers."

For example, to have print requests accepted for a class of printers named *class1*, enter:

```
/usr/lib/accept class1
```

If you want to prevent requests from being routed to a printer, you can use the **reject** command. The **-r** options allows you to send users a message explaining why a printer is out of service.

For example, to prevent printing requests, from being routed to a printer, *printer4*, because of repairs, enter:

```
/usr/lib/reject -r"printer4 needs repair" printer4
```

A user who requests a file to be printed on *printer4* will receive the following message:

```
lp:can not accept requests for destination "printer4"
–printer printer4 needs repair
```

To find out the acceptance status of printing destinations, enter:

```
lpstat -a
```

For more information on **lpstat**, see section 4.12.4, *Introduction to XENIX*. For more information on **accept/reject**, see **accept**(C), XENIX *Reference Manual*.

7.9.9 Enabling and Disabling Printers

The **enable** command allows **lpsched** to print files on printers. A printer can accept requests for printing after the **accept** command is given for it, but in order for the files to be printed, the **enable** command must be given as well.

For example, to enable a printer named *daisy*, enter:

```
enable daisy
```

You can disable printers with the **disable** command. The scheduler, **lpsched**, will not send printing requests to disabled printers regardless of their acceptance status. The **-r** options allows you to send a message to users explaining why a printer has been disabled.

For example, to disable a printer named *laser* because of a paper jam, enter:

```
disable -r"paper jam" laser
```

Users requesting the status of *laser* with the command **lpstat -plaser** will receive the following mesage:

```
printer "laser" disabled since Dec 5 10:15
paper jam
```

For more information on these two commands, see **enable**(C) and **disable**(C) in the XENIX *Reference Manual*.

7.9.10 Printer Interface Programs

Each printer on your system must have a printer interface program. This can be a shell script, C program, or any other executable program. Your XENIX system provides a model interface program. It is written as a shell script and can be found in */usr/spool/lp/model*. You can use this program as is, modify it, or write your own interface program.

If you want to write or modify a printer interface program, the following information may be helpful.

When **lpsched** routes a printing request to a printer *p*, */usr/spool/lp* invokes the interface program for *p* as follows:

interface *id user title copies options file*

with

interface	the directory which contains executable copies of interface programs
P	the interface program being executed
id	the request id returned by **lp**
user	the login name of the user who made the request
title	an optional title given by the user
copies	the number of copies requested
options	a list of printer dependent options separated by blanks
file	the full pathname of a file to be printed

When the interface program is started, its standard input comes from */dev/null* and both standard output and standard error output are directed to the printer's device. Devices are opened for reading as well as writing when file modes permit. If a device is a regular file, all output is appended to the end of that file.

Interface programs may format their output in any way. They must, however, ensure proper **stty** modes for terminal characteristics such as baud rate and output options. In a shell script interface, this means that printer's device must be open for reading – take the standard input for the **stty** command from the device.

The file */etc/default/lpd* contains a line "BANNERS=d" where *d* is the number of banner pages to be printed at the front of every printing request. Interface programs should examine this file and behave accordingly.

After printing is completed, the interface program should exit with a code showing that the print job was successful. Exit codes are interpreted by the printer scheduler, **lpsched**, as follows:

Exit Code	Meaning to **lpsched**
0	Print job was successful
1 to 127	**lpsched** found a problem while printing this particular request, for example, too many unprintable characters. This problem will not affect future printing requests. **lpsched** notifies users by mail that there was an error in printing the request.
greater than 127	These codes are reserved for internal use by **lpsched**. Interface programs must not exit with codes in this range.

Finally, when problems occur in printing that are likely to affect future printing requests, the printer interface program should disable printers so that print requests are not lost. When a busy printer is disabled, the interface program will be terminated with a signal 15 so that print requests are not lost.

For more information on printer interface programs, see **lpadmin**(C), XENIX *Reference Manual*.

7.10 Adding Additional Memory

You can improve system performance, to a point, by increasing the amount of internal memory.

To increase internal memory follow these steps:

1. Install additional an memory board according to manufacturers instructions.

2. Run memory diagnostics. Use diagnostic software provided with DOS or your computer hardware.

3. When you have the memory installed and confirmed, reboot the system.

4. The **boot** program reconfigures the system based on the amount of internal memory available. Check the messages displayed during the boot process to verify that the system is finding the additional memory. You should notice the increased memory available. The number of multiscreens, number of buffers, and message buffer size may also increased.

Note

Some machines have a hardware limitation on the maximum amount of memory that can be installed. Refer to your computer hardware manual to determine the maximum amount of memory you can install.

You can follow the same procedure if you wish remove internal memory from the system.

If the memory hardware reports an error to XENIX, XENIX displays the message:

panic: parity

You then see the software reboot message:

** Normal System Shutdown **

** Safe to Power Off **
-or-
** Press Any Key to Reboot **

If the system repeatedly panics from parity errors, consider replacing the memory chips.

7.11 Adding a Second Hard Disk

You can give the system extra room for storing users' files and directories by adding a second hard disk to the system. This is often the only remedy for a system that has one hard disk and suffers from chronic lack of space. See the *Release Notes* for a list of hard disks compatible with the current XENIX release.

You can only have one disk controller card and the ROM of both the disk controller card and the host board must support the appropriate configuration information in the ROM tables. In particular, the ROM tables need to support the number of heads and cylinders. Check your hardware manual for the hard disk drive and the computer for this information. Switch settings on the disk controller card may need to be changed.

Before adding the new disk, you must know how to connect it to the computer. Connecting the hard disk is explained in the hardware manual provided with the disk. Make sure the second drive passes the manufacturers diagnostics before running XENIX.

This is an outline of the procedure to add a second hard disk:

- Connect the hard disk, then boot the system and enter system maintenance mode.
- Use the **/etc/mkdev** program. **mkdev** executes **hdinit** which, in turn, runs three interactive programs: **fdisk**(C), to separate the disk into MS-DOS and XENIX areas (the DOS area is optional), **badtrk**(M), to identify and map any bad areas on your disk, and **divvy**(C), to partition your disk into file systems.

These are the steps to add a second hard disk with one XENIX file system and no DOS area:

1. Connect the hard disk, then boot the system and enter system maintenance mode.
2. When you are in system maintenance mode, enter:

   /etc/mkdev hd

3. After a while, you see the main **fdisk** menu:

   Select one of the following options or 'q' to exit the program

   1. Display Partition Table
   2. Use Entire Disk for XENIX
   3. Create XENIX Partition
   4. Activate XENIX Partition
   5. Delete XENIX Partition

   Please enter your choice:

If you want to use your whole disk for XENIX, enter '2', then press RETURN. **fdisk** displays a table in which the entire disk is allocated for XENIX.

Partition	Status	Type	Start	End	Size
1	Active	UNKNOWN	000	1223	1224

Warning! All data on your disk will be lost!
Do you wish to continue? (y/n)

If you would like XENIX to occupy the whole disk, enter 'y' and press RETURN.

You see the partition table again, with the following changes:

Partition	Status	Type	Start	End	Size
1	Active	XENIX	000	1223	1224

Total disk size: 1224

Press RETURN to continue

Press RETURN, and you see the main **fdisk** menu. The program then prompts for a RETURN to continue. After you press RETURN, you see the main **fdisk** menu, shown above. Type 'q' to leave **fdisk** and continue with the installation.

If you had a large portion of the disk already allocated to DOS, you must run DOS to deallocate this area. See **fdisk**(C) for more information on sharing disks between DOS and XENIX. No matter what configuration you produce with **fdisk**, the active partition must be the XENIX partition when you are through.

4. The next menu is from the bad track mapping program, **badtrk**. It displays this menu:

Select one of the following options, or 'q' to exit program:

1. Print Current Bad Track Table
2. Scan Media Surface for Possible Bad Spots
3. Create New Bad Track Table
4. Add Entries to Current Bad Track Table by Head/Cylinder #
5. Add Entries to Current Bad Track Table by Block Number
6. Delete Entries from Bad Track Table.

Please enter your choice:

Enter "2," then press RETURN. The program now scans the active partition of the new disk for flaws. The larger your disk, the longer the scanning process takes, so a very large disk may take a while. Common times are between one and two minutes per megabyte of storage.

As **badtrk** scans the disk, it displays the number of each sector it examines. As it discovers flaws, it prints their locations.

When the scan is complete, the main menu reappears. Enter "1" to see what flaws the scanning process disclosed. The program automatically enters any flaws it discovers in the new bad track table.

If your disk comes with a flaw map, cross–check your map against the results of the disk scanning. Because most disk flaws are marginal or intermittent, any flaw map will almost certainly contain more flaws than the scanning process will reveal. Depending on the format of your flaw map, select either option 4, Add Entries by Head/Cylinder #, or option 5, Add Entries by Block Number, and enter the flaws, one per line. Enter 'q' when you are finished.

Exit **badtrk** by entering 'q' at the main menu.

5. Next, you see a prompt from the file system partitioning program, **divvy**.

   Do you require block–by–block control over
   the layout of the XENIX partition? (y/n)

   If you answer "y", the entire second hard disk is made into a single filesystem, */u*, which can be used as a user filesystem. If you answer "n", you will see the main **divvy** menu. Refer to the **divvy**(C) manual page for information on using **divvy** to create multiple XENIX filesystems.

7.11.1 Mounting Another Filesystem

To use the second disk, or create a second, mounted filesystem, enter this command:

```
mkdev fs /dev/u /u
```

This command does the following:

- Creates a directory */u* (also known as the mount point).
- Creates the *lost+found* directory (used by **fsck** to recover files if the filesystem is corrupted).
- Mounts the device (*/dev/u*) on */u*.
- Removes write permissions on the directory */u* for group and all other users except root.

- Creates files in the */u/lost+found* directory, then removes them. This allocates inodes for the directory, so that if the filesystem is corrupted and runs out of inodes, **fsck**(C) is still able to recover files.

- Adds the following line to */etc/checklist*:

  /dev/u

- Modifies */etc/rc* so that the new filesystem is automatically mounted, checked and cleaned when appropriate.

7.11.2 Using A Second Filesystem

You can access the files on your new filesystem by first mounting it in the appropriate directory (we used */u* as an example in the preceeding section). After you mount the filesystem, all directories and files on it are usable just as any others on the system.

If you want to have user accounts on your mountable filesystem, change the variable "HOME" in the file */etc/default/mkuser*. The "HOME" variable defines the directory where user accounts are placed. This variable is used by the **mkuser**(C) program, which adds new users to the system.

Edit the file */etc/default/mkuser*. There is a line which may look like this:

HOME=/usr

Change */usr* to the name of the directory where you want to place user accounts. In our example, this is */u*. The line should now read:

HOME=/u

Now, whenever you run the **mkuser**(C) script to add a new user, that user's account will be in */u*.

If there are already users on the system, and you want to move their accounts to the new filesystem, you can use the **copy**(C) command to copy their accounts to the new filesystem. You must also change the users' entries in */etc/passwd* to reflect the new pathnames of their home directories.

Follow these steps to move user accounts from one filesystem to another:

1. Make sure the new filesystem is mounted. Also, be sure you are either in system maintenance mode, or logged in as root.

2. Change directories to the top of the current user account directory. If, for example, the user accounts are in */usr*, enter:

   ```
   cd /usr
   ```

   and press RETURN.

3. List the contents of this directory:

   ```
   lc
   ```

   You see a list of account names, for example:

   ```
   alisonb   dean    jerrys       sams
   blf       gregt   lost+found   steve
   buckm     jeffj   pj           vicki
   ```

4. Enter:

   ```
   copy -orm /usr /u
   ```

   and press RETURN.

5. When the **copy** command has finished, enter:

   ```
   cd /u
   ```

   and press RETURN. List the new contents of */u* to make sure all of the accounts have been copied correctly.

6. **After you are sure that all of the accounts have been completely copied**, you can remove the user accounts in the previous user filesystem.

7. Change the home directory for each user as listed in */etc/passwd*. An example entry in */etc/passwd* might be:

   ```
   alisonb:CoHiKNs.:271:104:Alison Berry:/usr/alisonb:/bin/csh
   ```

   You see one such line for every user on your system. Change the field:

:/usr/alisonb:

to match the user's new home directory:

:/u/alisonb:

Do this for every user who's home directory has changed.

Chapter 8

Solving System Problems

8.1 Introduction

This chapter explains how to solve problems that affect the operation of the system. The problems range in complexity from how to fix a nonechoing terminal, to how to restore lost system files.

8.2 Restoring a Nonechoing Terminal

A nonechoing terminal is any terminal that does not display characters entered at the keyboard. This abnormal operation can occur whenever a program stops prematurely as a result of an error, or when the user presses the BREAK key.

To restore the terminal to normal operation, follow these steps:

1. Press the Ctrl-J key. The system may display an error message. If it does, ignore the message.

2. Enter:

   stty sane

   and press the Ctrl-J key. The terminal does not display what you enter, so enter accurately.

After pressing Ctrl-J, the terminal should be restored and you may continue your work.

8.3 Solving Lineprinter Problems

No printing can be done on the lineprinter spooling system unless the print scheduler, **lpsched**, is running. To check the status of **lpsched**, enter:

```
lpstat -r
```

To restart **lpsched**, enter the following on two lines:

```
rm -f /usr/spool/lp/SCHEDLOCK
/usr/lib/lpsched
```

The lock file */usr/spool/lp/SCHEDLOCK* prevents more than one version of **lpsched** from running at any one time.

Access to files and directories in */usr/spool/lp* by **lp** can be another source of spooling problems. You can check the **lpsched** log file, */usr/spool/lp/log*. This is a record of the print scheduler's activity and

errors. If **lpsched** refuses to run or a printer refuses to print, check to make sure that:

- The printer is enabled; see *lp*(C) in the XENIX *Reference Manual*.
- The files and directories in */usr/spool/lp* are readable and writable by **lp**.

For more information on the lineprinter spooling system, see the section "Adding A Lineprinter" in Chapter 7.

8.4 Stopping a Runaway Process

A runaway process is a program that cannot be stopped from the terminal at which it was invoked. This occurs whenever an error in the program "locks up" the terminal, that is, prevents anything you enter from reaching the system.

To stop a runaway process, follow these steps:

1. Go to a terminal that is not locked up.
2. Log in as the super-user.
3. Enter:

   ps -a

   and press the RETURN key. The system displays all current processes and their process identification numbers (PIDs). Find the PID of the runaway program.

4. Enter:

   kill *PID*

   and press the RETURN key. The *PID* is the process identification number of the runaway program. The program should stop in a few seconds. If the process does not stop, enter:

   kill -9 *PID*

   and press the RETURN key.

The last step is sure to stop the process, but may leave temporary files or a nonechoing terminal. To restore the terminal to normal operation, follow the instructions in the section "Restoring a Nonechoing Terminal" in this chapter.

8.5 Replacing a Forgotten Password

The XENIX operating system does not provide a way to decipher an existing password. If a user forgets his password, the system manager must change the password to a new one. To change an ordinary user password, follow the instructions in the section "Changing a User's Password" in Chapter 3.

8.6 Removing Hidden Files

A hidden file is any file whose name begins with a dot (.). You can list the hidden files in a directory by entering:

```
lc -a
```

and pressing the RETURN key.
You can remove most hidden files from a directory by entering:

```
rm .[a-z]*
```

and pressing the RETURN key. Remaining files can be removed individually.

8.7 Restoring Free Space

The system displays an "out of space" message whenever the root directory has little or no space left to work. To restore system operation, you must delete one or more files from the root directory. To delete files, follow the steps outlined in the section "Maintaining Free Space" in Chapter 5.

8.8 Restoring Lost System Files

If a system program or data file is accidentally modified or removed from the file system, you can recover the file from the periodic backup disk with the *sysadmin* program. To restore the files, follow the instructions in the section "Restoring a Backup File" in Chapter 6.

8.9 Restoring an Inoperable System

On very rare occasions, one or more of the critical XENIX system files may be accidentally modified or removed, preventing the system from operating. In such a case, you must reinstall the XENIX system, and restore user program and data files from backup disks. To reinstall the system, follow the instructions in the XENIX *Installation Guide*. To restore files from backup disks, follow the instructions in the section "Restoring a Backup File" in Chapter 6.

8.10 Recovering from a System Crash

A system crash is a sudden and dramatic disruption of system operation that stops all work on the computer. System crashes occur very rarely. They are usually the result of hardware errors or damage to the root file system which the operating system cannot correct by itself. When a system crash occurs, the system usually displays a message explaining the cause of the error, then stops. This gives the system manager the chance to recover from the crash by correcting the error (if possible), and restarting the system.

A system crash has occurred if the system displays a message beginning with "panic:" on the system console, or the system refuses to process all input (including INTERRUPT and QUIT keys) from the system console and all other terminals.

To recover from a system crash, follow these steps:

1. Use the error message(s) displayed on the system console to determine the error that caused the crash. If there is no message, skip to step 3.

2. Correct the error, if possible. A complete list of error messages and descriptions for correcting the errors is given in *messages*(M) in the XENIX *Reference Manual*. (Even if the problem cannot be located or corrected, it is generally worthwhile to try to restart the system at least once by completing the remaining steps in this procedure.)

3. Turn off the computer and follow the steps described in Chapter 2, "Starting the System," to restart the system.

4. If the system will not restart, or crashes each time it is started, the operating system is inoperable and must be reinstalled. Follow the procedures described in the XENIX *Installation Guide* to reinstall the system and in Chapter 6, "Backing Up File Systems," to restore user's files.

5. If the system cannot be started from the "Boot" disk in the distribution set for installation, the computer has a serious hardware malfunction. Contact a hardware service representative for help.

8.11 Mapping a Bad Track

Bad tracks on the hard disk are mapped during the XENIX installation procedure. This allows XENIX to avoid those areas of the disk that cannot be read or written. However, hard disks can develope bad tracks after XENIX is installed and running. If this occurs, the **badtrk**(M) utility should be run by the super-user (root) enabling XENIX to avoid the new bad track(s). **badtrk** must be run in single-user mode. Use **/etc/shutdown su** to enter single-user mode from multi-user mode (see **shutdown**(C)).

badtrk is a menu-driven utility for viewing, adding, or deleting entries to the bad track table. See **badtrk**(M) in the XENIX *Reference Manual* for more on its options and their use.

8.12 Changing XENIX Initialization

One common problem is adapting the system initialization to suit your system environment. This problem occurs whenever you have added new devices such as terminals or disk drives to the system, and wish these devices to be automatically enabled or mounted whenever you start normal system operation. You can adapt system initialization by modifying the system initialization files.

The XENIX initialization files contain XENIX commands and/or data which the system reads at system startup or whenever a user logs in. The files typically mount file systems, start programs, and set home directories and terminal types. The initialization files are named */etc/rc*, *.profile*, and */etc/motd*.

The system manager may modify these files to create any desired initial environment. The files are ordinary text files and may be modified using a text editor such as **ed** (see the XENIX *User's Guide*). Note, however, that the */etc/rc* and *.profile* files contain XENIX commands and comments, and have the command file format described in Chapter 4, "The Shell," in the XENIX *User's Guide*.

8.12.1 Changing the */etc/rc* File

The */etc/rc* file contains XENIX system initialization commands. The system executes the commands at system startup. The commands display a startup message, start various system daemons, and mount file systems. You can display the contents of the file with the **more** command. Enter:

more /etc/rc

and press the RETURN key.

You may change the contents of the file so that the system executes any set of commands you wish. For example, if you want the system to automatically mount a new file system, simply append the appropriate **mount** command in the file. The system will execute the command on each startup.

To append a command to the file, follow these steps:

1. Log in as the super-user.

2. Invoke a text editor and specify the */etc/rc* as the file to be edited.

3. Locate the place in the file you wish to insert the command (e.g., if the command mounts a file system, insert it with other mounting commands).

4. Insert the command on a new line. Make sure you enter the command correctly. The system rejects any incorrect commands and the commands that follow it when the file is read at system startup.

5. Exit the editor.

No other changes to the file are required. Be careful not to delete any commands already in the file unless you are sure they are not needed.

8.12.2 Changing the *.profile* Files

The *.profile* files contain commands that initialize the environment for each user. The commands in the file are executed whenever the user logs in. The file usually contains commands that set and export various system variables (e.g., TERM, PATH, MAIL). These variables give the system information such as what terminal type is being used, where to look for programs the user runs, where to look for the user's mailbox, what keys to expect for the "kill" and "backspace" functions, and so on (see Chapter 4, "The Shell," in the XENIX *User's Guide*).

There is one *.profile* file for each user account on the system. The files are placed in the user's home directory when the account is created. An ordinary user may modify his own *.profile* file or allow the system manager to make modifications. In either case, the file can be edited like the */etc/rc* file by using a text editor. Commands can be added or removed as desired.

8.12.3 Changing the */etc/motd* File

The message of the day file, */etc/motd*, contains the greeting displayed whenever a user logs in. Initially, this file contains the name and version number of the XENIX system. It can be modified to include messages such as a reminder to clean up directories, a notice of the next periodic backup, and so on.

The */etc/motd* file is an ordinary text file, so you can change the message by editing the file with a text editor. One common change is to include a reminder to delete unused files in order to preserve disk space. In general, you should limit the size of the file to include no more than a screenful of information.

Chapter 9

Building a Micnet Network

9.1 Introduction

A Micnet network allows communications between two or more independent XENIX systems. The network consists of computers connected by serial communication lines (that is, RS-232 ports connected by cable). Each computer in the network runs as an independent system, but allows users to communicate with the other computers in the network through the **mail**, **rcp**, and **remote** commands. These commands pass information such as mail, files, and even other commands, from one computer to another.

It is the system manager's task to build and maintain a Micnet network. The system manager decides how the computers are to be connected, makes the actual physical connections, then uses the **netutil** program to define and start the network.

This chapter explains how to plan a network and then build it with the **netutil** program. In particular, it describes:

- How to choose machine names and aliases
- How to draw the network topology map
- How to assign serial lines
- How to create the Micnet files
- How to distribute the Micnet files
- How to test the Micnet network

9.2 Planning a Network

To build a Micnet network, the **netutil** program requires that you provide the names of the computers that will be in the network, a description of how the computers are to be connected, a list of the serial lines to be used, the names of the users who will use the network, and what aliases (if any) they will be known by.

To keep the task as simple as possible, you should take some time to plan the network and make lists of the information you will be required to supply. To help you make these lists, the following sections suggest ways to plan a network.

9.2.1 Choosing Machine Names

A Micnet network requires that each computer in the network have a unique "machine name." A machine name helps distinguish each computer from other computers in the network. It is best to choose machine names as the first step in planning the network. This prevents confusion later on, when you build the network with the **netutil** program.

A machine name should suggest the location of the computer or the people who use it. You can also use any name you wish. The name must be unique and consist of letters and digits. The Micnet programs use only the first eight characters of each name, so be sure those characters are unique.

The **netutil** program saves the machine name of a computer in a */etc/systemid* file. One file is created for each computer. After you have built and installed the network, you can find out the machine name of the computer you are using by displaying the contents of this file.

9.2.2 Choosing a Network Topology

The network topology is a description of how the computers in the network are connected. In any Micnet network, there are two general topologies from which all topologies can be constructed. These are "star" and "serial."

In a star topology, all computers are directly connected to a central computer. All communications pass through the central computer to the desired destination.

In a serial topology, the computers form a chain, with each computer directly connected to no more than two others. All communications pass down the chain to the desired destination.

A network may be strictly star, strictly serial, or a combination of star and serial topologies. The only restriction is that no network may form a ring. For example, you cannot close up a serial network by connecting the two computers at each end.

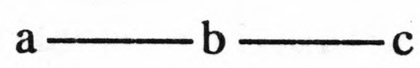
The kind of topology you choose depends on the number of computers you have to connect, how quickly you want communications to proceed, and how you want to distribute the task of passing along communications. A star topology provides fast communication between computers, but requires both a large portion of the central computer's total operation time and a large number of serial lines on the central computer. A serial topology distributes the communication burden evenly, requiring only two serial lines per computer, but is slow if the chain is very long (communication between computers can take several minutes). Often a combination of star and serial topologies makes the best network. In any

case, make the choice you think is best. If you discover you have made a wrong choice, you may change the network at any time.

9.2.3 Drawing a Network Topology Map

A network topology map is a sketch of the connections between computers in the network. You use the map to plan the number and location of the serial lines used to make the network.

You can make the map while you work out the topology. Simply arrange the machine names of each computer in the network on paper, then mark each pair of computers you wish to connect with serial lines. For example, the topology map for three computers might look like this:



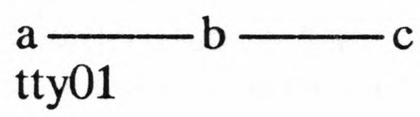
As you draw, make sure that there is no more than one connection between any two computers in the network. Furthermore, make sure that no rings are formed (a ring is a series of connections that form a closed circle). Multiple connections and rings are not permitted.

9.2.4 Assigning Lines and Speeds

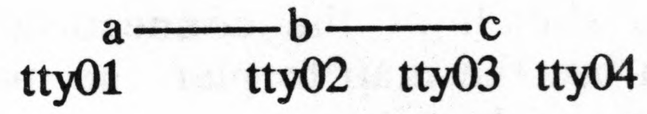
Once you have made the topology map, you can decide which serial lines to use. Since every connection between computers in the network requires exactly two serial lines (one on each computer), you need to be very careful about assigning the lines. Follow these steps:

1. Make a list of the serial lines (TTY lines) available for use on each computer in the network. You can display a list of the serial lines on a computer by displaying the file */etc/ttys*. A line is available if it is not connected to any device such as a terminal or modem.

2. Using the topology map, first pick a computer, then assign one (and only one) serial line to each connection shown for that computer. The serial lines must be from the list of available lines for that computer. No line may be assigned more than once. For example, if computer *a* has only one available serial line (tty01), then the topology map should look like this:



3. Repeat step 2 for all computers in the topology map. Make sure that each connection is assigned a line and that no two connections on any given computer have the same line. When finished, the map should look like this:



If a computer does not have enough available serial lines to meet its needs, you can make the lines available by removing the devices already connected to them. If you cannot remove devices, you must redraw your topology map.

4. Using the topology map, assign a serial line transmission speed for each computer pair. The speed must be within the normal range for XENIX serial lines (typically 110 to 9600). Transmission speeds are a matter of preference. In general, a higher speed means a smaller amount of time to complete a transmission, but a greater demand on system's input and output capabilities. In some cases, transmission speeds are a matter of hardware capabilities. Some hardware is not capable of transmission speeds greater than 1200 baud. For this reason, 1200 is the recommended speed when first installing Micnet. You may then increase the speed if you find the hardware can support it.

5. After the topology map is completely filled in, make a list of all computer pairs, showing their machine names, serial lines, and transmission speeds. You will use this list when installing the network.

9.2.5 Choosing Aliases

Once you have decided how to connect the computers in the network, you can choose aliases for users in the network. An alias is a simple name that represents both a location (computer) and a user. Aliases are used by the **mail** command to allow you to refer to specific computers and users in a network without giving the explicit machine and user names. Although not a required part of the network, aliases can make the network easier to use and maintain.

There are three kinds of aliases: standard, machine, and forward. A standard alias is a name for a single user or a group of users. A machine alias is a name for a computer or an entire network (called a site). A forward alias is a temporary alias for a single user or group of users. A forward alias allows users who normally receive network communications at one computer to receive them at another.

When you build a network with the **netutil** program, you are asked to provide standard aliases only. (You can incorporate machine and forward aliases into the network at your leisure.) Each standard alias must have a unique name and a list of the login names of the users it represents. You may choose any name you wish as long as it consists of letters and numbers, begins with a letter, and does not have the same spelling as the login names. The name should suggest the user or group of users it represents. The login names must be the valid login names of users in the network.

To help you prepare the aliases for entry during the **netutil** program, follow these steps:

1. Make a list of the user aliases (that is, the aliases that refer to just one user) and the corresponding login names of each user.

2. Make a separate list of the group aliases (that is, the aliases that refer to two or more users) and the login names or user aliases (from the first list) of the corresponding users. A group alias may have any number of corresponding users.

Note that there are a number of predefined group aliases. The name **all** is the predefined alias for all users in the network. The machine names of the computers in the network are predefined aliases for the users on each computer. Do not use these names when defining your own aliases.

9.3 Building a Network

You build a network with the **netutil** program. The program allows you to define the machines, users, and serial lines that make up the network.

To build a network, you must first create the Micnet files that define the network and then transfer these files to each computer in the network. After each computer receives the files, you may start the network and use it to communicate between computers.

The following sections describe how to build the network.

9.3.1 Creating the Micnet Files

The Micnet files are created with the **install** option of the **netutil** program. The **install** option asks for the names, aliases, and serial lines of each computer in the network. As you supply the information, it automatically creates the files needed for each computer. These files can then be transferred to the other computers in the network with the **save** and **restore** options of **netutil**. This means you can build the entire network from just one computer.

To use the **install** option, follow these steps:

1. Log in as the super-user.

2. Enter:

   netutil

   and press the RETURN key. The program displays the network utility menu. The **install** option is the first item in the menu.

3. Enter the number 1, and press the RETURN key. The program displays the following message:

   Compiling new network topology
   Overwrite existing network files? (yes/no)?

   Enter *y* and press the RETURN key to overwrite the files. The existing network files must be overwritten to create the new network. The first time you install the network, these files contain default information that need not be saved. If you install the system a second time or expand the system, it may be wise to save a copy of these files before starting the **install** option. The files can be saved on a floppy or a hard disk with the **save** option described later in this chapter.

   Once you have entered *y*, the program displays the following message:

   Enter the name of each machine
   (or press RETURN to continue installation).
   Machine name:

4. Enter the machine name and press the RETURN key. You may enter more than one name on a line by separating each with a comma or a space. After you have entered all the names, press the RETURN key to continue to the next step. The program displays the names you entered and asks if you wish to make changes.

5. Enter *y* (for "yes") if you wish to enter all the names again. Otherwise, enter *n* (for "no") or just press the RETURN key to move on to the next step. If you enter *n*, the program displays the message:

   For each machine, enter the names of the machines
   to be connected with it
   Machine a:
   Connect to:

6. Using the list of machine pairs you created when planning the network, enter the machine names of the computers connected to the given computer. You may enter more than one name on a line by separating each name with a comma (,) or a space. When you have entered the machine names of all computers connected to the given computer, press the RETURN key. The program prompts for the names of the computers connected to the next computer.

7. Repeat step 5 for all remaining computers. As the program prompts for each new set of connections, it will show a list of the machine names it already knows to be connected with the current computer. You need not enter these names. The program automatically checks for loops. If it finds one, it ignores the machine name that creates the loop and prompts for another.

   Finally, when you have given the connections for all computers in the network, the program displays a list of the connections and asks if you wish to make corrections.

8. Enter *y*, if you wish to enter the connections again. Otherwise, enter *n*, to move to the next step. If you enter *n*, the program displays the message:

   ```
   For each machine pair, enter the tty name and tty speeds
   For the a <==> b machine pair.
    Tty on a:
   ```

9. Using the list of serial line assignments you created when planning the network, enter the serial line name or number (for example, tty03 or 3) for the first computer in the pair and press the RETURN key. The program displays the message:

   ```
   Tty on b:
   ```

10. Enter the serial line name for the second computer in the pair and press the RETURN key. The program displays the message:

    ```
    Speed:
    ```

11. Enter the speed (for example, 1200) and press the RETURN key. The program asks for the serial lines and transmission speed of the next pair.

12. Repeat step 8 for all remaining machine pairs. When you have given serial lines and speeds for all pairs, the program displays this information and asks if you wish to make corrections.

13. Enter *y*, if you wish to enter the serial lines and speeds again. Otherwise, enter *n*, to move to the next step.

The program displays the message:

Enter the names of users on each machine:

For machine a:
Users on a:

14. Enter the login name of a user on the given computer, then press the RETURN key. You may enter more than one name on a line by separating each name with a comma (,) or a space. When you have entered all names for the given computer, press the RETURN key. The program displays the names of the users on the computer and asks if you wish to make corrections.

15. Enter *y*, if you wish to enter the user names again. Otherwise, enter *n*. If you enter *n*, the program prompts you for the names of the users on the next computer.

16. Repeat steps 13 and 14 for all remaining computers. After you have entered the names of users for every computer, the program prompts you to enter any aliases:

Do you wish to enter any aliases? (yes/no)?

17. Enter *y*, if you wish to enter aliases. Otherwise, enter *n*, to complete the installation. If you enter *y*, the program displays the message:

Each alias consists of two parts, the first is the alias name,
the second is a list of one or more of the following:
valid user names
previously defined aliases
machine names

Aliases:

18. Using the list of aliases you created when planning the network, enter the name of an alias and press the RETURN key. The program displays the message:

Users/Aliases:

19. If the alias is to name a single user, enter the login name of that user and press the RETURN key. The program then prompts for another alias.

If, on the other hand, the alias is to name several users, enter the login names of the users. If one or more of the users to be named by the alias are already named by other aliases, enter the aliases instead of the login names. If all the users on one computer are to be named by the alias, enter the machine name instead of the login names. In

any case, make sure that each item entered on the line is separated from the next by a comma (,) or a space. If there are more items than can fit on the line, enter a comma after the last item on that line and press the RETURN key. You can then continue on the next line. After all names and aliases have been entered, press the RETURN key. The program then prompts you for another alias.

20. Repeat steps 17 and 18 for all remaining user aliases in your list. When you have given all aliases, press the RETURN key. The program displays a list of all aliases and their users and asks if you wish to make corrections.

21. Enter *y*, if you wish to enter all aliases again. Otherwise, enter *n*, to complete the installation.

Once you direct **netutil** to complete the installation, it copies the information you have supplied to the network files, displaying the name of each file as it is updated. Once the files are updated, you may use the **save** option to copy the Micnet files to floppy disk.

9.3.2 Saving the Micnet Files

You can save copies of the Micnet files on backup media (floppy disk) or hard disk with the **save** option of the **netutil** program. Saving the files allows you to transfer them to the other computers in the network. Before you can save the files to a floppy you need to format a floppy disk (see the section "Formatting Floppy Disks" in Chapter 4). Saving the files to the hard disk enables you to use **uucp**(C) to transfer the files to other machines.

To save the files, follow these steps:

1. Log in as the super-user.

2. Enter:

   netutil

   Press the RETURN key. The program displays the network utility menu.

3. Enter the number 2, and press the RETURN key. The program displays the message:

   Save to /dev/fd*x* (yes/no)?

   where *x* is a drive number.

4. If you wish to use the specified disk drive, insert a blank, formatted floppy disk into the drive, wait for the drive to accept the disk, then enter "yes", and press the RETURN key. If you do not wish to use the drive, enter "no", and press the RETURN key. The program displays a prompt asking you for the filename of the disk drive (or file) you wish to use. Insert a blank, formatted disk into your chosen drive, wait for the drive to accept the disk, then enter the filename of the drive. The name of the default backup device (disk drive) is specified in the file */etc/default/micnet*. This device can be changed depending on system configuration.

   In either case, the program copies the Micnet files to the floppy disk.

5. Remove the floppy disk from the drive. Using a soft tip marker (do not use a ball point pen), label the disk "Micnet disk".

As soon as all files have been copied, you can transfer them to all computers in the network.

9.3.3 Restoring Micnet Files

The last step in building a Micnet network is to copy the Micnet files from the Micnet disk to all computers in the network. Do this with the **restore** option of the **netutil** program. For each computer in the network, follow these steps:

1. Log in as the super-user.

2. Enter:

   netutil

   Press the RETURN key. The program displays the network utility menu.

3. Enter the number 3, and press the RETURN key. The program displays the message:

   Restore from /dev/fd*x* (yes/no)?

   where *x* is the number of a drive.

4. If you wish to use the specified disk drive, insert the Micnet disk into the drive, wait for the drive to accept the disk, then enter "yes" and press the RETURN key. If you do not wish to use the drive, enter "no" and press the RETURN key. The program displays a prompt asking you for the filename of the disk drive you wish to use. Insert the Micnet disk into your chosen drive, wait for the drive to accept the disk, then enter the filename of the drive.

   In either case, the program copies the network files to the appropriate directories, displaying the name of each file as it is copied. Finally, the program displays the message:

   Enter the name of this machine:

5. Enter the machine name of the computer you are using and press the RETURN key. The program copies this name to the new */etc/systemid* file for the computer. If necessary, it also disables the serial lines to be used on the computer, preparing them for use with the network.

When the files have been copied, you may start the network with the **start** option.

9.4 Starting the Network

Once the Micnet files have been transferred to a computer, you can start the network with the **start** option of the **netutil** program. The **start** option starts the Micnet programs which perform the tasks needed to communicate between the computers in the network.

To start the network, follow these steps for each computer in the network:

1. Log in as the super-user.

2. Enter:

   netutil

   Press the RETURN key. The system displays the network utility menu.

3. Enter 4, and press the RETURN key. The program searches for the */etc/systemid* file. If it finds the file it starts the network. If it does not, it prompts you to enter the machine name of the computer and then creates the file. The program also asks if you wish to log errors and transmissions. In general, these are not required except when checking or testing the network. When starting the network for the first time, enter *n* in response to each question and press the RETURN key.

Once the network has started, you may move to the next computer and start the network there.

Note that, for convenience, you can let each computer start the network automatically whenever the system itself is started. Simply include the command:

```
netutil start
```

in the system initialization file, */etc/rc*, of each computer. To add this command, use a text editor as described in the section "Changing the /etc/rc File" in Chapter 8. You can add the **-x** or **-e** options to this command line if you wish to log transmissions or errors. Even if you do not use these options, Micnet copies a log in and log out message to the system *LOG* file each time you start and stop the network. This means you will need to periodically clear the file. See the section "Clearing Log Files" in Chapter 5.

9.5 Testing a Micnet Network

After you have started a network for the first time, you should test the network to see that it is properly installed. In particular, you must determine whether or not each computer is connected to the network.

To test the network, you will need to know how to use the **mail** command (see Chapter 6, "Mail," in the XENIX *User's Guide*). The following sections explain how to test the network and how to correct the network if problems are discovered.

9.5.1 Checking the Network Connections

You can make sure that all computers are connected to the network by mailing a short message to **all** (the alias for all users in the network) with the **mail** command. Follow these steps:

1. Choose a computer.

2. Log in as the super-user.

3. Use the **mail** command (see the XENIX *User's Guide*) and the **all** alias to mail the message:

```
Micnet test
```

to all users in the network.

4. Check the mailboxes of each user in the network to see if the message was received. To check the mailboxes, log in as the super-user at each computer and use the **cat** command to display the contents of each user's mailbox.

   The name of each user's mailbox has the form:

   /usr/spool/mail/*login-name*

   where *login-name* is the user's login name.

The network is properly installed when all users have received the message. If the users at one or more computers fail to receive the message, the computers are not properly connected to the network. To fix the problem, you need to locate the computer which has failed to make a connection. The next section explains how to do this.

9.5.2 Using the LOG File to Locate a Problem

You can locate a problem with connections by examining the *LOG* files on each computer in the network. The *LOG* files contain records of the interaction between each pair of computers. There are two *LOG* files for each pair of computers (one file on each computer). The *LOG* files on any given computer are kept in subdirectories of the */usr/spool/micnet* directory. Each subdirectory has as its name the *machine-name* of the other computer in the pair. You can examine the contents of a *LOG* file by entering:

cat /usr/spool/micnet/remote/*machine-name*/LOG

and pressing the RETURN key. The *machine-name* must be the name of a computer that is paired with the computer you are using.

Each *LOG* file should contain a "startup message" which lists the name of each computer in the pair, and the serial line through which the pair is connected. It also shows the date and time at which the network was started. The message should look like:

```
daemon.mn: running as MASTER
Local system: a
Remote system: b, /dev/tty02
Tue Sep 24 22:30:35 1985
```

A startup message is added to the file each time the network starts successfully. If the message is not present, one or more of the the network files and directories cannot be found. Make sure that you have used the **restore** option to transfer all the network files to the computer. Also, make sure that the */etc/systemid* file contains the correct machine name for the given computer.

Each *LOG* file will contain a "handshake" message if the connection between the computer pair has been established. The message:

first handshake complete

is added to the file on a successful connection. If the message is not present, make sure that the network has been started on the other computer in the pair. The network must be started on both computers before any connection can be made. If the network is started on both computers but the handshake message does not appear, then the serial line may be damaged or improperly connected. Check the serial line to make sure that the cable is firmly seated and attached to the correct RS-232 connectors on both computers. If necessary, replace the cable with one known to work.

If both the startup and handshake messages appear in the *LOG* file but the network is still not working, then there is a problem in transmission. You can create a record of the transmissions and errors encountered while transmitting by restarting the network and requesting Micnet to log all transmissions and errors. Just enter *y* (for "yes") when the **start** option asks if you wish to log errors or transmissions.

Error entries contain the error messages generated during transmission. Each message lists the cause of the error and the subroutine which detected the error. For example, the message:

rsync: bad Probe resp: 68

shows that the *rsync* subroutine received a bad response (character 68 hexadecimal) from the other computer. You may use this information to track down the cause of the problem. One common problem is stray information being passed down the serial line by electronic noise. Make sure that the serial line's cable is properly protected against noise (for example, that the cable does not lie near any electric motor, generator, or other source of electromagnetic radiation). Also make sure the cable is in good condition.

Transmission entries contain a record of normal transmissions between computers. Each entry lists the direction, byte count, elasped time, and time of day of the transmission. For example, the entry:

rx: 0c 01 22:33:49

shows that 12 characters (0c hexadecimal) were received (*rx*) at 22:33:49. The elasped time for the transmission was 1 second. You can use the records to see if messages are actually being transmitted.

9.5.3 Stopping the Network

You can stop the network with the **stop** option of the **netutil** program. This option stops the Micnet programs, stopping communication between computers in the network.

To stop the network, follow these steps on each computer in the network:

1. Log in as the super-user.
2. Enter:

   netutil

   Press the RETURN key. The program displays the network utility menu.

3. Enter 5, and press the RETURN key. The program stops the network programs running on the computer.

9.5.4 Modifying the Micnet Network

You can modify a Micnet network at any time by changing one or more of the Micnet files. You can reinstall the network with the **netutil** program. For very small changes (for example, correcting the spelling of an alias), you can modify the Micnet files directly with a text editor. The files and their contents are described in detail in the M section of the XENIX *Reference Manual*.

Before making any changes to a file, a copy should be made. You can make a copy with the **cp** command. You can replace an old file with the updated file using the **mv** command. Once one or more files have been changed on one computer, the files must be transferred to the other systems in the network using the **save** and **restore** options. These options can only be used after you have stopped the network.

Note that changes to the *aliases* file will not be incorporated into the system until the **aliashash** program is executed. This program produces the *aliases.hash* file needed by the network to resolve aliases. See *aliashash*(M) in the XENIX *Reference Manual* for a description of this command.

9.6 Using a Uucp System

You can send and receive mail from other Micnet sites by installing a **uucp** system on one computer in your site. A **uucp** system is a set of XENIX programs that provide communication between computers using ordinary telephone lines.

To use a **uucp** system with your Micnet network, follow these steps:

1. Install a **uucp** system on one computer in the Micnet site. Installation of a uucp system requires a modem and the uucp software provided with the XENIX *Operating System*. See the XENIX *User's Guide* for complete details.

2. Add the entry:

   uucp:

   to the *maliases* file of the computer on which the uucp system is installed.

3. For all other computers in your site, add the entry:

   uucp:*machine-name*:

   to the *maliases* file. The *machine-name* must be the name of the computer on which the **uucp** system is installed. One may also use the longer form of entry on the computer on which the uucp system is installed.

You can test the **uucp** system by mailing a short letter to yourself via another site. For example, if you are on the site "chicago", and there is another Micnet site named "seattle" in the system, then the command:

```
mail seattle!chicago!johnd
```

will send mail to the "seattle" site, then back to your "chicago" site, and finally to the user "johnd" in your Micnet network. Note that a **uucp** system usually performs its communication tasks according to a fixed schedule, and may not return mail immediately.

Chapter 10

Installing Device Drivers

10.1 Introduction

This chapter explains how to install device drivers in order to support additional peripherals. It describes how to use the facilities of XENIX to install new device driver modules. After a device driver is written and compiled (or purchased) it is necessary to relink the XENIX kernel, including the new device driver, allowing XENIX to make use of the device. The tools necessary to write and compile a device driver are included with the XENIX Development System. However, the Link Kit is included with the XENIX Operating System in case you acquire a device driver you wish to incorporate into your kernel.

10.1.1 What is a XENIX Device Driver?

For each peripheral device in a XENIX system, there must be a "device driver" to provide the software interface between the device and the system. A XENIX device driver is a set of routines that communicates with a hardware device, and provides a means by which XENIX can control the device in order to perform Input/Output (I/O) operations.

10.1.2 Why Do You Install A Device Driver?

A device driver will normally be supplied as a single software module, contained in a single file, which must be installed on the XENIX System before the device can be used. The installation of this software is as necessary an operation as the actual hardware installation. It must be completed before the device can be used.

10.1.3 What You Need To Know

In order to install a new device driver you should:

- First install the hardware device on your system according to the manufacturer's instructions.
- Be able to edit data files using one of the XENIX text editors.
- Be familiar with the format of the file *c.c*, *c.o*, and the driver files which are located in the */usr/sys/conf* directory on your system.
- Install the Link Kit on your system.

The precise instructions for installing a new device driver will be different for each type of device. In this chapter, we use examples which should be supplemented by the specific installation instructions that will have been provided with the device driver software.

10.1.4 The Examples In This Chapter

Throughout this chapter we will use two examples to illustrate the process of installing a device driver. We assume that your system has already been in use and that you have recently acquired a new high quality printer and a new high capacity disk drive, each of which you wish to install on your system. With each of these devices a single diskette has been supplied which contains the necessary software to enable you to complete the installation.

The examples assume that you have installed the hardware on your system according to the manufacturer's instructions and that you now wish to install the software so you can use the new peripheral devices.

10.2 Installing Device Drivers

In order to install the device driver for a new peripheral you must boot the system and enter system maintenance mode. All the operations described as part of the installation process are carried out in this mode.

10.2.1 Where The Software Is Installed

Some software vendors may provide automatic installation utilities compatible with the standard XENIX System V installation utilities. In this case, entering the command:

```
# /etc/install
```

is sufficient to insure proper installation of the new device driver software and linking of a new version of the XENIX kernel including the new device driver.

However, all drivers may not be provided with such utilities. In these cases, follow the detailed instructions in the Section "First Steps In Installing New Device Drivers" and subsequent sections.

Installable device drivers have a directory reserved for their use. Most of the files associated with these device drivers are located in this directory. The directory is */usr/sys/conf* and contains the software modules which comprise the installable device drivers, together with one file *c.c* which is used to describe the configuration of your system.

In addition to the files in the */usr/sys/conf* directory you will need to add new files to the */dev* directory.

10.2.2 First Steps In Installing New Device Drivers

To begin the installation of the new device drivers you will need to to be certain the Link Kit is installed on your system. In order to save disk space, you should only install the Link Kit when you intend to use it, and remove the files when you are finished. The Link Kit may be installed (or removed) at any time by using the **custom**(C) utility. Follow the detailed instructions which have been supplied with the product. These instructions will usually tell you to copy the device driver module from the diskette supplied with the product into the */usr/sys/conf* directory and to make certain changes to the */usr/sys/conf/c.c* file. You should check the current contents of the */usr/sys/conf* directory to make sure that you will not overwrite an existing file by doing this. If there is an existing file of the same name as the one you wish to install, then you must rename the existing file in order to preserve it.

Some device drivers are provided in the standard configuration. When installing one of these device drivers, you don't need to modify or replace *c.c*.

Copying the files into the */usr/sys/conf* directory is normally done using the **tar** utility. A command such as:

- tar tf /dev/rfd0

will display the contents of the diskette. In our example we have two diskettes, one supplied with the high quality printer and one supplied with the high capacity disk drive.

Each of these diskettes contains the following:

- tar tf /dev/rfd0
  /usr/sys/conf/c.o
  /usr/sys/conf/hqp.o

- tar tf /dev/rfd0
  /usr/sys/conf/c.o
  /usr/sys/conf/hcd.o

Clearly, we have a name clash with the *c.o* file, there is one on each diskette and an existing one in the */usr/sys/conf* directory.

The following commands show how to temporarily save the existing *c.o* file, read in each of the supplied files from the diskettes, and save the *c.o* file supplied with each product as *c.hqp* for the printer and *c.hcd* for the disk:

- cd /usr/sys/conf
- mv c.o Mc.o
- : Insert the diskette containing the printer software
- tar xf /dev/rfd0
- mv c.o c.hqp
- : Insert the diskette containing the disk software
- tar xf /dev/rfd0
- mv c.o c.hcd
- mv Mc.o c.o

10.2.3 Editing the link_xenix File

The next step in installing the device drivers is to edit the file */usr/sys/conf/link_xenix* to include the description of the new configuration devices and their device drivers. This shell script is included with the Link Kit which links the kernel. This is provided for those users who don't have the XENIX Development System and therefore cannot use the *makefile* provided. You must add the name of the new driver module to the **ld** line, just before any **-l** flags.

The shell script tests for the existence of a *xenix* file, backs it up if it exists, links a new *xenix* and makes a *namelist* file (see **nm**(CP)). You do not need to know all the details of the specifications included in this file, since the information will be supplied to you with the installation instructions for the device.

In cases where the driver is not part of the standard configuration file and no replacement for the standard file *c.o* is provided, you need to modify the configuration file *c.c* and recompile the file. You must have the XENIX Development System installed on your system in order to do this.

In most cases, it will be sufficient for you to simply copy the device driver specification line and declarations from the *c.c* file, supplied with the device, into the *c.c* file describing your system. Use **diff**(C) to locate what needs to be added. Make a backup copy of the *c.c* file before making modifications. You must be careful to check for clashes between existing devices and new devices. These clashes can occur in any one of three of the fields in a device driver specification - the *major device number* for either the *block* or *character* device fields, and the interrupt vector numbers used by the device.

The rules for resolving these clashes are fairly simple. If an existing device has the same *major device number* as that given for the new device, then simply choose a new, unique, number for the new device and use that as the *major device number*. This rule applies to both the *block* and *character* device fields. Note that if you have to change the *major device number* from that supplied in your installation instructions, then you must use this new number when creating the *special files* described in a later section. If you wish to replace an existing device with the new device, then you should choose the same *major device number* as the existing device. Try to choose the same number for both block and character devices.

The new device driver must use an interrupt vector not currently in use by another driver. This can be determined by examining the appropriate tables in *c.c*. Also, depending on which interrupt vectors are used, the system interrupt priority masks in the file *primask.c* may need modification. Be sure to back up the original file before making modifications.

Note

Errors in configuring the system interrupt priority mask can result in the creation of a kernel which will not run. Exercise extreme caution when modifying this file. You should only modify this file if you understand the XENIX System software and hardware interrupt priority schemes.

The file *primask.c* contains two arrays. One defines the interrupt vectors masked out by each interrupt vector, and the other defines the interrupt vectors masked by the various Software System Priority Levels (spl's). First you must decide which vector you will use; then you must decide which system priority level your driver will use. This is determined at the time the driver is written and is normally System Priority Level 5 (spl5) or a lower priority level. The mask bit corresponding to the hardware vector the device will use must be set in the System Priority Level Mask at the chosen priority level and all higher levels. All the masks for hardware devices which interrupt at the same, or higher, priority must have the same mask bit set.

When the necessary changes to *c.c* and *primask.c* are made, you must compile the files using the *makefile* provided. Use the *link_xenix* shell script, as described above, to link a new kernel.

10.2.4 Creating Special Device Files

In order for programs to gain access to the newly installed devices they must also exist as files within the filesystem. These files are termed *special files* and are generally all located in the */dev* directory. Once again, the specific installation instructions supplied with the device will give the precise details of the name to be used for the *special file* and the other parameters associated with it. In order to create a *special file* you use the **mknod** command. You must supply the name of the special file, its type (which can be either a *block* device or a *character* device) and the *major* and *minor* device numbers associated with the device. For example:

- cd /dev
- /etc/mknod hcd0 b 1 0
- /etc/mknod rhcd0 c 1 0
- /etc/mknod hqp c 7 0

Note that we have used the normal XENIX convention for setting up the disk device names. A digit has been appended to the mnemonic to indicate the drive number, and the "raw" device (that is the *character special* device) name has been created with an "r" prefix.

10.2.5 How To Boot The System

Test the new kernel before installing it as */xenix*. To do so, enter the following:

```
# cp /usr/sys/conf/xenix /xenix.new
# /etc/haltsys
```

The system now reboots. The computer prompts:

```
Boot
:
```

By pressing the *RETURN* key, or simply doing nothing, the default operating system image */xenix* will be loaded and started. In order for the bootstrap program to locate and load the newly installed device drivers, it must be told to read the */xenix.new* file which contains the device driver. Enter:

```
xenix.new
```

and press RETURN. The system is now running with the "new" kernel. Test the various devices (especially the one(s) added).

ps(C) will not work correctly unless you specify the **-n** flag and the pathname of the *xenix* kernel you are using.

Note

Do not install *xenix* on the hard disk, as */xenix*, until it is fully tested.

10.2.6 Creating A New /xenix

When the kernel is satisfactory, install the new kernel on the hard disk. Enter the following:

```
# cd /usr/sys/conf
# ./hdinstall
```

hdinstall backs up the "old" */xenix* and copies */usr/sys/conf/xenix* to */xenix*.

Appendix A
XENIX Special Device Files

A.1 Introduction

This appendix contains information needed to create file systems and add terminals to the XENIX system. For a full description of the special files mentioned here, see section M–HW of the XENIX *Reference Manual* found at the back of this volume.

A.2 File System Requirements

Many of the file system maintenance tasks described in this guide require the use of special filenames, block sizes, and gap and block numbers. The following sections describe each in detail.

A.3 Special Filenames

A special filename is the name of the device special block or character I/O file, which corresponds to a peripheral device, such as a hard or floppy disk drive. These names are required in such commands as **mkfs**, **mount**, and **df** to specify the device containing the file system to be created, mounted, or searched.

The following table lists the special filenames and corresponding devices, for hard and floppy disk drives on a typical computer.

Device Special Filenames – Disks

Filename	Disk Drive
/dev/fd0	Floppy Drive 0
/dev/fd1	Floppy Drive 1
/dev/hd00	Entire hard disk
/dev/root	Root file system
/dev/usr	User file system

A.4 Block Sizes

The block size of a disk is the number of blocks of storage space available on the disk, where a block is typically 512 or 1024 bytes of storage. Refer to the **machine**(M) manual page or use **cmchk**(C) to determine the size of blocks on your system. Many commands require input that defines the number of blocks to be operated on. Other commands report disk space in terms of 512 byte blocks, in particular **df**, **du**, **ls**, **lc**, and **find**. A 500 byte file on a 1024 byte block file system is reported as using 2 blocks by these utilities, as the file uses one system block which is equivalent to two 512

byte blocks. The size of a 10 megabyte hard disk in 1024 byte blocks is 9792. Note that some of the blocks on the disk are reserved for system use and cannot be accessed by user programs. The block size of a typical floppy disk depends on the total storage capacity of the disk, as given by the manufacturer.

A.5 Gap and Block Numbers

The gap and block numbers are used by the **mkfs**, and possibly **fsck**, commands to describe how the blocks are to be arranged on a disk. The following table lists the gap and block numbers for the floppy and hard disks used with a typical computer.

Disks	Gap	Block
Floppy Disk, 48ds9	1	9
Floppy Disk, 96ds15	1	15
Hard Disk	1	34

The number of blocks can also be determined by multiplying the number of sectors per track (usually 17) by the number of heads on the hard disk.

A.6 Terminal and Network Requirements

The **enable** and **disable** commands are used to add and remove terminals on a system. The install option of the **netutil** program is used to build a network. The preceding commands and option require the names of the serial lines through which a terminal or network is to be connected. The following table lists the device special filenames of the two serial lines (actually two serial ports either with or without modem control). The character I/O files correpsonding to these serial lines can be found in the */dev* directory. Note that the files */dev/console* and */dev/tty02* through */dev/tty10* represent "hardwired" devices and are not available for connection to terminals or hardware. Also, refer to **serial**(M) for more information on serial lines.

Filename	Line
/dev/tty11 or /dev/tty1a	main serial line (without modem control)
/dev/tty12 or /dev/tty2a	alternate serial line (without modem control)
/dev/tty13 or /dev/tty1A	main serial line (with modem control)
/dev/tty14 or /dev/tty2A	alternate serial line (with modem control)

Appendix B

XENIX Directories

B.1 Introduction

This appendix lists the most frequently used files and directories in the XENIX system. Many of these files and directories are required for proper XENIX operation and must not be removed or modified. The following sections briefly describe each directory.

B.2 The Root Directory

The root directory (/) contains the following system directories:

/bin	XENIX command directory
/dev	Device special directory
/etc	Additional program and data file directory
/lib	C program library directory
/mnt	Mount directory (reserved for mounted file systems)
/usr	User home directories
/tmp	Temporary directory (reserved for temporary files created by programs)

All of the above directories are required for system operation.

The root directory also contains a few ordinary files. Of these files, the most notable is the *xenix* file which contains the XENIX kernel image.

B.3 The */bin* Directory

The */bin* directory contains the most common XENIX commands, that is, the commands likely to be used by anyone on the system. The following is a list of a few of the commands:

basename	echo	passwd	su
cp	expr	rm	sync
date	fsck	sh	tar
dump	login	sleep	restor
dumpdir	mv	stty	test

These commands and all others in the /bin directory are required.

B.4 The */dev* Directory

The */dev* directory contains special device files which control access to peripheral devices. All files in this directory are required, and must not be removed.

The following is a list of the files:

/dev/console	System console
/dev/lp	Lineprinter
/dev/mem	Physical memory
/dev/null	Null device (used to redirect unwanted output)
/dev/rXX	Unbuffered interface to corresponding device name
/dev/root	Root file structure
/dev/swap	Swap area
/dev/ttyXX	Terminals
/dev/tty	The terminal you are using

B.5 The */etc* Directory

The */etc* directory contains miscellaneous system program and data files. All files are required, but many may be modified.

The following program and data files must not be removed or modified:

/etc/mnttab	Mounted device table
/etc/mount	For mounting a file structure
/etc/mkfs	For creating a file structure
/etc/init	First process after boot

The following data files may be modified, if desired. No files may be removed.

/etc/passwd	Password file
/etc/rc	Bootup shell script
/etc/ttys	Terminal set up
/etc/termcap	Terminal capability map
/etc/motd	Message of the day

The data files in the directory */etc/default* contain default information which is used by system commands (see **default**(M)). The following data files may be modified. No files may be removed.

/etc/default/backup	backup(C) default information
/etc/default/console/keys	mapkey(M) default information
/etc/default/console/screen	mapkey(M) default information
/etc/default/console/strings	mapkey(M) default information
/etc/default/dumpdir	dumpdir(C) default information
/etc/default/login	login(M) default information
/etc/default/lpd	lp(C) default information
/etc/default/micnet	micnet(M) default information
/etc/default/mkuser	mkuser(C) default information
/etc/default/passwd	passwd(C) default information
/etc/default/quot	quot(C) default information
/etc/default/restor	restore(C) default information
/etc/default/su	
/etc/default/tar	tar(C) default information

B.6 The */lib* Directory

The */lib* directory contains runtime library files for C and other language programs. The directory is required.

B.7 The */mnt* Directory

The */mnt* directory is an empty directory reserved for mounting removable file systems.

B.8 The */tmp* Directory

The */tmp* directory contains temporary files created by XENIX programs. The files are normally present when the corresponding program is running, but may also be left in the directory if the program is prematurely stopped. You may remove any temporary file that does not belong to a running program.

B.9 The */usr* Directory

The */usr* directory contains the home directories of all users on the system. It also contains several other directories which provide additional XENIX commands and data files.

The */usr/bin* directory contains more XENIX commands. These commands are less frequently used or considered nonessential to XENIX system operation.

The */usr/include* directory contains header files for compiling C programs.

The */usr/lib* directory contains more libraries and data files used by various XENIX commands.

The */usr/spool* directory contains various directories for storing files to be printed, mailed, or passed through networks.

The */usr/tmp* directory contains more temporary files.

The */usr/adm* directory contains data files associated with system administration and accounting. In particular, the */usr/adm/messages* file contains a record of all error messages sent to the system console. This file is especially useful for locating hardware problems. For example, an unusual number of disk errors on a drive indicates a defective or misaligned drive. Since messages in the file can accumulate rapidly, the file must be deleted periodically.

B.10 Log Files

A variety of directories contain log files that grow in size during the normal course of system operation. Many of these files must be periodically cleared to prevent them from taking up valuable disk space (see the section "Clearing Log Files" in Chapter 5). The following table lists the files (by full pathname) and their contents.

Filename **Description**

/etc/ddate
: Records date of each backup.

/usr/adm/pacct
: Records accounting information; grows rapidly when process accounting is on.

/usr/adm/messages
: Records error messages generated by the system when started.

/usr/adm/wtmp
: Records user logins and logouts.

/usr/adm/sulog
: Records each use of the **su** command; grows only if option is set in the */etc/default/su* file.

/usr/spool/at/past
: Records each use of the **at** command.

/usr/spool/micnet/remote/*/LOG
: Records transmissions between machines in a Micnet network. The * must be the name of a remote machine connected to the current machine.

Index

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

W

X